OpenShift Sandboxed Containers 1.5

OpenShift Sandboxed Containers リリースノート

OpenShift Container Platform の場合

Last Updated: 2024-04-11

OpenShift Sandboxed Containers 1.5 OpenShift Sandboxed Containers リリースノート

OpenShift Container Platform の場合

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 3
多様性を受け入れるオープンソースの強化 3

第1章 概要 4

第2章 OPENSHIFT SANDBOXED CONTAINERS 1.5 リリースノート 5
2.1. このリリースについて 5
2.2. 新機能および機能拡張 5
2.3. バグ修正 6
2.4. 既知の問題 6
2.5. 非同期エラータの更新 9

はじめに

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。この取り組みは膨大な作業を要するため、これらの変更による更新は可能な範囲で段階的に行われます。詳細は、弊社 の CTO、Chris Wright のメッセージ を参照してください。

第1章 概要

第2章 OPENSHIFT SANDBOXED CONTAINERS 1.5 リリースノート

2.1. このリリースについて

これらのリリースノートは、OpenShift Container Platform 4.15 とともに OpenShift サンドボックスコンテナー 1.5 の開発を追跡します。

OpenShift Container Platform は FIPS 用に設計されています。FIPS モードでブートされた Red Hat Enterprise Linux (RHEL) または Red Hat Enterprise Linux CoreOS (RHCOS) を実行する場合、OpenShift Container Platform コアコンポーネントは、**x86_64**、**ppc64le**、および **s390x** アーキテクチャーのみで、FIPS 140-2/140-3 検証のために NIST に提出された RHEL 暗号化ライブラリーを使用します。

NIST 検証プログラムの詳細は、暗号化モジュール検証プログラム を参照してください。検証のために提出された RHEL 暗号化ライブラリーの個別バージョンの最新の NIST ステータスについては、政府の標準規格 を参照してください。

2.2. 新機能および機能拡張

2.2.1. AWS および Azure の柔軟な Pod VM インスタンスサイズ

OpenShift Sandboxed Containers 1.5 からは、Pod VM のインスタンスサイズを指定できるようになりました。AWS の場合は **PODVM_INSTANCE_TYPES** フィールド、Azure の場合は、**peer-pods-cm ConfigMap** CR の **AZURE_INSTANCE_SIZES** を使用できます。詳細は、Web コンソールを使用した AWS 用のピア Pod ConfigMap の作成 および Web コンソールを使用した Azure 用のピア Pod ConfigMap の作成 を参照してください。

2.2.2. AWS および Azure での Pod VM イメージの自動作成

OpenShift Sandboxed Containers 1.5 からは、**peer-pods-secret** オブジェクトと **peer-pods-cm** オブジェクトが存在し、または変数の値が空の場合は、**peer-pods-cm** に **AZURE_IMAGE_ID** または **PODVM_AMI_ID** 変数が含まれません。手順の詳細は、Web コンソールでの KataConfig カスタムリソースの作成 を参照してください。

2.2.3. 管理者が kata ノードのインストール、アンインストール、および更新の操作をより深く理解できるようになります。

kataNodes という名前の新しいフィールドが導入され、**kata** 操作を受けているノードの状態のより詳細なビューがユーザーに表示されます。既存の **Is In Progress** ブール値ステータスフィールドは、より有益な **InProgress** 条件に置き換えられました。

詳細は、移行のインストールとアンインストール を参照してください。

2.2.4. IBM Z および IBM(R) LinuxONE 上の OpenShift Sandboxed Containers のピア Pod のサポート (テクノロジープレビュー)

ユーザーは、IBM Z および IBM® LinuxONE (390x アーキテクチャー) 上のピア 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。

2.3. バグ修正

- OpenShift Sandboxed Containers 1.5.0 から 1.5.2 では、ユーザーがピア Pod を作成すると、Go 1.21.1 でのネットワーク機能の動作の変更により、Pod は **ContainerCreating** 状態のままになり、"Failed to identify the host primary interface" エラーが表示されました。この問題は、OpenShift Sandboxed Containers 1.5.3 で修正されています。(KATA-2847)

- 以前は、**KataConfig** CR のインストール中にその削除を開始すると、OpenShift Sandboxed Containers Operator がどちらのプロセスも完了せずに、削除とインストールを同時に試行していました。このリリースでは、Operator はインストールの完了後に削除が実行されるようにシリアル化します。(KATA-1851)

- 以前のリリースでは、具体的にラベル付けされたノードを使用してデプロイされた kata 対応クラスターを更新できませんでした。ノードラベルに変更を加えても、デプロイメントの変更がトリガーされませんでした。既存の **kataConfig** CR を削除し、更新されたラベルで新しい **kataConfig** CR を作成する必要がありました。以前のリリース (リリース 1.4) から、ノードラベルを更新すると、デプロイメントの変更が自動的にトリガーされます。(KATA-1928)

- 以前は、QEMU が **virtiofsd** を検出していないと、**kata** ワークロードが削除されるたびに、QEMU はシステムジャーナルにエラーを記録していました。このリリースでは、**kata** ランタイムは **virtiofsd** を停止する前に QEMU を停止するようになりました。この修正は、OpenShift Container Platform 4.13 および 4.14 でのみ利用可能です。(KATA-2133)

- 以前は、**KataConfig** CR でピア Pod を有効にし、インストール後に CR を検査すると、 **kata-remote** ランタイムクラスが **status.runtimeClass** フィールドに表示されませんでした。この問題は、OpenShift Sandboxed Containers 1.5.0 で修正されています。(KATA-2164)

- 以前は、ピア Pod VM の実行中に **peerpodconfig-ctrl-caa-daemon** Pod を再起動すると、同じピア Pod を表す複数の VM が作成される場合がありました。クラウドプロバイダーのコンソールまたは CLI からインスタンスを手動で削除しない限り、冗長インスタンスは、元のピア Pod が実行している限り存在していました。この更新により、**peerpodconfig-ctrl-caa-daemon** Pod を再起動した後、新しいピア Pod VM が作成され、古いインスタンスはすぐに削除されます。(KATA-2519)

- 以前のリリースでは、AWS または Azure で実行されているピア Pod VM のインスタンスメタデータを要求すると、AWS または Azure Instance Metadata Service は Pod ではなくワーカーノードのメタデータを返していました。リリース 1.5.1 の更新により、AWS または Azure Instance Metadata Service は予想通りに Pod のメタデータを返します。(KATA-2583)

2.4. 既知の問題

- OpenShift Container 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 拒否を受け取る場合があります。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アノテーションを追加する必要があります。

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

(KATA-1376)

- 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、Pod ログに次のエラーが表示されます。

```
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)

- 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)

- **--announce-submounts** または **--thread-pool-size** のいずれかを含む **io.katacontainers.config.hypervisor.virtio_fs_extra_args** アノテーションを持つ Pod は起動しません。これは、OpenShift Container Platform 4.13 および 4.14 上の OpenShift Sandboxed Containers Operator によって使用される **virtiofsd** コンポーネントの回帰です。OpenShift Container Platform 4.12 および 4.11 は影響を受けません。(KATA-2146)

- 一時メモリーボリュームの **sizeLimit** オプションは、OpenShift Sandboxed Containers では機

能しません。一時ボリュームサイズのデフォルトは、サンドボックスコンテナーに割り当てられたメモリーの 50% です。ボリュームを再マウントすることで、このボリュームのサイズを手動で変更できます。たとえば、サンドボックスコンテナーに割り当てられたメモリーが 6 GB で、一時ボリュームが **/var/lib/containers** にマウントされている場合は、次のコマンドを使用して、このボリュームのサイズを仮想マシンメモリーのデフォルトの 50% を超えて増やすことができます。。

```
$ mount -o remount,size=4G /var/lib/containers
```

(KATA-2579)

- **io.katacontainers.config.hypervisor.default_vcpus** アノテーションおよび **io.katacontainers.config.hypervisor.default_memory** アノテーションは QEMU のセマンティクスに従いますが、ピア Pod には次の制限があります。
  - **io.katacontainers.config.hypervisor.default_memory** アノテーションの値を **256** 未満に設定すると、次のエラーが発生します。

    ```
    Failed to create pod sandbox: rpc error: code = Unknown desc = CreateContainer failed:
    Memory specified in annotation io.katacontainers.config.hypervisor.default_memory is
    less than minimum required 256, please specify a larger value: unknown
    ```

  - **io.katacontainers.config.hypervisor.default_memory: 256** アノテーションおよび **io.katacontainers.config.hypervisor.default_vcpus: 1** アノテーションを使用する場合は、リストから最小のインスタンスが起動されます。
  - **io.katacontainers.config.hypervisor.default_vcpus: 0** アノテーションを使用する場合は、すべてのアノテーションが無視され、デフォルトのインスタンスが起動します。

  代わりに、柔軟な Pod VM サイズには **io.katacontainers.config.hypervisor.machine_type: <instance type/instance size>** アノテーションを使用することが推奨されます。(KATA-2575、KATA-2577、KATA-2578)

- OpenShift Sandboxed Containers Operator 1.4.1 からバージョン 1.5 への自動アップグレード中に、アップグレードが **pending** 状態でスタックします。
  サブスクリプションが自動更新に設定されていると、OpenShift Sandboxed Containers のアップグレードがインストールされます。ただし、**KataConfig** CR (カスタムリソース) がインストールされている場合、CSV は **pending** 状態のままになります。

  次のコマンドを実行して、**Subscription** オブジェクトのステータスを確認できます。

  ```
  $ oc get sub osc-operator -n openshift-osc-operator -o yaml
  ```

  次のエラーが、**Subscription** オブジェクトの **status** セクションと、アップグレードの **InstallPlan** オブジェクトの **status** セクションに表示されます。

  ```
  message: 'error validating existing CRs against new CRD"s schema for
  "kataconfigs.kataconfiguration.openshift.io":
      error validating custom resource against new schema for KataConfig /example-
  kataconfig:
      [].status.runtimeClass: Invalid value: "string": status.runtimeClass in body
      must be of type array: "string"'
  ```

  このエラーが発生した場合は、OpenShift Sandboxed Containers Operator をアンインストールしてから再インストールする必要があります。

1. **kata** ランタイムまたは **kata-remote** ランタイムで実行しているワークロード (Pod、デプロイメント、デーモンセット) をすべて削除します。これらのワークロードは、再インストール後に再作成する必要があります。ワークロードの削除の詳細は、CLI を使用した OpenShift Sandboxed Containers Pod の削除 を参照してください。

2. **KataConfig** CR を削除します。CLI を使用した KataConfig カスタムリソースの削除 を参照してください。

重要

ワークロードが実行中の場合は、**KataConfig** CR を削除しないでください。

次のコマンドを使用して、**KataConfig** CR の削除ステータスを確認できます。

```
$ oc get kataconfig -n openshift-osc-operator
```

3. Operator をアンインストールします。CLI を使用したSandboxed Containers Operator のインストール を参照してください。

4. OpenShift Sandboxed Containers Operator を再インストールします。CLI を使用した Sandboxed Containers Operator のインストール を参照してください。
OpenShift Sandboxed Containers Operator の再インストールはバージョン 1.5.0 をインストールします。

5. **KataConfig** CR を作成します。CLI を使用した KataConfig カスタムリソースの作成 を参照してください。

6. ワークロードを再作成します。CLI を使用した Sandboxed Containers へのワークロードのデプロイ を参照してください。

注記

サブスクリプションを手動更新に設定している場合は、OpenShift Sandboxed Containers Operator 1.5.1 が利用可能になるまでアップグレードを承認しないでください。

(KATA-2593)

2.5. 非同期エラータの更新

OpenShift Sandboxed Containers 4.15 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 OpenShift Container Platform 4.15 エラータは、Red Hat カスタマーポータルから入手できます。非同期エラータについては、OpenShift Container Platform 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することができます。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

OpenShift Container Platform のエラータ通知メールを生成させるには、Red Hat カスタマーポータルのユーザーアカウントでシステムが登録されており、OpenShift Container Platform 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 1.5 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。

2.5.1. RHEA-2023:7493 - OpenShift Sandboxed Containers 1.5.0 イメージのリリース、バグ修正、機能強化のアドバイザリー

発行日: 2023 年 11 月 27 日

OpenShift Sandboxed Containers リリース 1.5.0 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一覧は、RHEA-2023:7493 アドバイザリーに記載されています。

2.5.2. RHBA-2024:0147 - OpenShift Sandboxed Containers 1.5.1 イメージのリリースとバグ修正アドバイザリー

発行日: 2024 年 1 月 11 日

OpenShift Sandboxed Containers リリース 1.5.1 が利用可能になりました。このアドバイザリーには、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-2024:0147 アドバイザリーに記載されています。

2.5.3. RHBA-2024:0815 - OpenShift Sandboxed Containers 1.5.2 イメージのリリースとバグ修正アドバイザリー

発行日: 2024 年 2 月 15 日

OpenShift Sandboxed Containers リリース 1.5.2 が利用可能になりました。このアドバイザリーには、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、 the RHBA-2024:0815 アドバイザリーに記載されています。