OpenShift Sandboxed Containers 1.5

OpenShift sandboxed containers release notes

For OpenShift Container Platform

Last Updated: 2024-03-05

OpenShift Sandboxed Containers 1.5 OpenShift sandboxed containers release notes

For OpenShift Container Platform

Abstract

The release notes summarize all new features and enhancements, notable technical changes, major corrections from the previous version, and any known bugs upon general availability.

Table of Contents

PREFACE

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. Because of the enormity of this endeavor, these changes are being updated gradually and where possible. For more details, see our CTO Chris Wright's message .

CHAPTER 1. INTRODUCTION

CHAPTER 2. OPENSHIFT SANDBOXED CONTAINERS 1.5 RELEASE NOTES

2.1. ABOUT THIS RELEASE

These release notes track the development of OpenShift sandboxed containers 1.5 alongside OpenShift Container Platform 4.15.

OpenShift Container Platform is designed for FIPS. When running Red Hat Enterprise Linux (RHEL) or Red Hat Enterprise Linux CoreOS (RHCOS) booted in FIPS mode, OpenShift Container Platform core components use the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the **x86_64**, **ppc64le**, and **s390x** architectures.

For more information about the NIST validation program, see Cryptographic Module Validation Program. For the latest NIST status for the individual versions of RHEL cryptographic libraries that have been submitted for validation, see Compliance Activities and Government Standards.

2.2. NEW FEATURES AND ENHANCEMENTS

2.2.1. Flexible pod VM instance sizes for AWS and Azure

From OpenShift sandboxed containers 1.5, you can specify the instance size of a pod VM. You can use the **PODVM_INSTANCE_TYPES** field for AWS, or the **AZURE_INSTANCE_SIZES** for Azure in the **peer-pods-cm ConfigMap** CR. For more information, see Creating a peer-pod ConfigMap for AWS using the web console and Creating a peer-pod ConfigMap for Azure using the web console .

2.2.2. Automatic pod VM image creation on AWS and Azure

From OpenShift sandboxed containers 1.5, the pod VM images are automatically created if the **peer-pods-secret** and **peer-pods-cm** objects exist, and the **peer-pods-cm** does not contain the **AZURE_IMAGE_ID** or **PODVM_AMI_ID** variables or if the variable's value is empty. For more information about the procedure, see Creating the KataConfig custom resource in the web console

2.2.3. Empowering administrators with greater insight into the kata node install, uninstall and update operations

A new field named **kataNodes** has been introduced, presenting users with a more detailed view of the state of the nodes undergoing **kata** operations. The existing **Is In Progress**boolean status field has been replaced with a more informative **InProgress** condition.

For more information, see Installation and uninstall transitions.

2.2.4. Peer pods support for OpenShift sandboxed containers on IBM Z and IBM(R) LinuxONE (Technology Preview)

Users can now deploy OpenShift sandboxed containers workloads using peer pods on IBM Z and IBM® LinuxONE (390x architecture). This enables users to circumvent the need for nested virtualization. This feature is in Technology Preview and not fully supported. For more information, see Deploying OpenShift sandboxed containers workloads using peer pods.

2.3. BUG FIXES

- Previously, initiating the deletion of the **KataConfig** CR during its installation caused the OpenShift sandboxed containers Operator to attempt to delete and install simultaneously, without ever completing either process. With this release, the Operator serializes deletion to happen after the installation is complete. (KATA-1851)

- Previously, users could not update kata-enabled clusters that were deployed with specifically labeled nodes. Changes to node labels did not trigger deployment changes. Users had to delete the existing **kataConfig** CR and create a new **kataConfig** CR with the updated labels. Starting from the previous release (release 1.4), updating node labels automatically triggers a deployment change. (KATA-1928)

- Previously, when QEMU did not detect **virtiofsd**, QEMU logged errors in the system journal each time a **kata** workload was deleted. With this release, the **kata** runtime now stops QEMU before stopping **virtiofsd**. This fix is available for OpenShift Container Platform 4.13 and 4.14 only. (KATA-2133)

- Previously, when you enabled peer pods in the **KataConfig** CR and then examined the CR after installation, the **kata-remote** runtime class was not displayed in the **status.runtimeClass** field. This is fixed in OpenShift sandboxed containers 1.5.0. (KATA-2164)

- Previously, restarting **peerpodconfig-ctrl-caa-daemon** pods while peer-pod VMs were running may resulted in creating multiple VMs that represented the same peer pod. The redundant instances existed as long as the original peer pod was still running, unless you manually deleted the instances from the cloud provider console or CLI. With this update, after restarting a **peerpodconfig-ctrl-caa-daemon** pod, a new peer-pod VM is created and old instances are deleted immediately. (KATA-2519)

- Previously, when users requested the instance metadata of a peer pod VM running on AWS or Azure, the AWS or Azure Instance Metadata Service returned the metadata of the worker node instead of the pod. With the update for release 1.5.1, the AWS or Azure Instance Metadata Service returns the metadata of the pod, as expected. (KATA-2583)

2.4. KNOWN ISSUES

- You might receive SELinux denials when accessing files or directories mounted from the **hostPath** volume in a OpenShift Container Platform cluster. These denials can occur even when running privileged sandboxed containers because privileged sandboxed containers do not disable SELinux checks.
  Following SELinux policy on the host guarantees full isolation of the host file system from the sandboxed workload by default. This also provides stronger protection against potential security flaws in the **virtiofsd** daemon or QEMU.

  If the mounted files or directories do not have specific SELinux requirements on the host, you can use local persistent volumes as an alternative. Files are automatically relabeled to **container_file_t**, following the SELinux policy for container runtimes. See Persistent storage using local volumes.

  Automatic relabeling is not an option when mounted files or directories are expected to have specific SELinux labels on the host. Instead, you can set custom SELinux rules on the host to allow the **virtiofsd** daemon to access these specific labels. (KATA-469)

- Some OpenShift sandboxed containers Operator pods use container CPU resource limits to increase the number of available CPUs for the pod. These pods might receive fewer CPUs than requested. If the functionality is available inside the container, you can diagnose CPU resource issues by using **oc rsh <pod>** to access a pod and running the **lscpu** command:

```
$ lscpu
```

Example output

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

The list of offline CPUs will likely change unpredictably from run to run.

As a workaround, you can use a pod annotation to request additional CPUs rather than setting a CPU limit. CPU requests that use pod annotation are not affected by this issue, because the processor allocation method is different. Rather than setting a CPU limit, the following annotation must be added to the metadata of the pod:

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

(KATA-1376)

- When you set SELinux Multi-Category Security (MCS) labels in the security context of a container, the pod does not start and the following error is displayed in the pod log:

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

  The runtime does not have access to the security context of the containers when the sandboxed container is created. This means that **virtiofsd** does not run with the appropriate SELinux label and cannot access host files for the container. As a result, you cannot rely on MCS labels to isolate files in the sandboxed container on a per-container basis. This means that all containers can access all files within the sandboxed container. Currently, there is no workaround for this issue.

  (KATA-1875)

- FIPS compliance for OpenShift sandboxed containers only applies to the **kata** runtime class. The new peer pods runtime class **kata-remote** is not yet fully supported, and has not been tested for FIPS compliance. (KATA-2166)

- A pod with an **io.katacontainers.config.hypervisor.virtio_fs_extra_args** annotation that contains either **--announce-submounts** or **--thread-pool-size** does not start. This is a regression of the **virtiofsd** component used by the OpenShift sandboxed containers Operator on OpenShift Container Platform 4.13 and 4.14. OpenShift Container Platform 4.12 and 4.11 are not affected. (KATA-2146)

- The **sizeLimit** option for ephemeral memory volumes does not work with OpenShift sandboxed containers. The ephemeral volume size defaults to 50% of the memory assigned to the sandboxed container. It is possible to manually change the size of this volume by remounting the volume. For example, if the memory assigned to the sandboxed container is 6 GB and the ephemeral volume is mounted to **/var/lib/containers**, you can increase the size of this volume beyond the default 50% of the VM memory by using the following command:

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

(KATA-2579)

- The **io.katacontainers.config.hypervisor.default_vcpus** and **io.katacontainers.config.hypervisor.default_memory** annotations follow the semantics for QEMU, which has the following limitations for peer pods:

  - If you set the value of the **io.katacontainers.config.hypervisor.default_memory** annotation to less than **256**, you get the following error:

    ```
    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
    ```

  - If you use the **io.katacontainers.config.hypervisor.default_memory: 256** and **io.katacontainers.config.hypervisor.default_vcpus: 1** annotations, the smallest instance is launched from the list.

  - If you use the **io.katacontainers.config.hypervisor.default_vcpus: 0** annotation, all annotations are ignored and the default instance is launched.

  Instead, it is recommended to use the **io.katacontainers.config.hypervisor.machine_type: <instance type/instance size>** annotation for flexible pod VM sizes. (KATA-2575, KATA-2577, KATA-2578)

- During automatic upgrades from OpenShift sandboxed containers Operator 1.4.1 to version 1.5, the upgrade gets stuck in **pending** state.
  If your subscription is set to automatic updates, then the upgrade for OpenShift sandboxed containers is installed. However, if a **KataConfig** CR (custom resource) is installed, then the CSV gets stuck in the **pending** state.

  You can check the status of your **Subscription** object by running the following command:

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

  The following error appears in the **status** section of the **Subscription** object, and in the **status** section of the upgrade **InstallPlan** object:

  ```
  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"'
  ```

  If you receive this error, you must uninstall, and then reinstall the OpenShift sandboxed containers Operator:

  1. Delete any workloads (pods, deployments, daemonsets) running in either the **kata** or **kata-remote** runtimes. These workloads will need to be recreated after reinstalling. For more information about deleting workloads, see Deleting OpenShift sandboxed containers pods using the CLI.

  2. Delete the **KataConfig** CR. See Deleting the KataConfig custom resource using the CLI .

IMPORTANT

Do not delete the **KataConfig** CR if a workload is running.

You can check the deletion status of the **KataConfig** CR using the following command:

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

3. Uninstall the Operator. See Deleting the OpenShift sandboxed containers Operator using the CLI

4. Reinstall the OpenShift sandboxed containers Operator. See Installing the OpenShift sandboxed containers Operator using the CLI.
   Reinstalling the OpenShift sandboxed containers Operator installs version 1.5.0.

5. Create your **KataConfig** CR. See Creating the KataConfig custom resource using the CLI .

6. Recreate your workloads. See Deploying a workload in a sandboxed container using the CLI .

NOTE

If you set your subscription to manual updates, do not approve the upgrade until OpenShift sandboxed containers Operator 1.5.1 is available.

(KATA-2593)

2.5. ASYNCHRONOUS ERRATA UPDATES

Security, bug fix, and enhancement updates for OpenShift sandboxed containers 4.15 are released as asynchronous errata through the Red Hat Network. All OpenShift Container Platform 4.15 errata is available on the Red Hat Customer Portal . See the OpenShift Container Platform Life Cycle for more information about asynchronous errata.

Red Hat Customer Portal users can enable errata notifications in the account settings for Red Hat Subscription Management (RHSM). When errata notifications are enabled, users are notified via email whenever new errata relevant to their registered systems are released.

NOTE

Red Hat Customer Portal user accounts must have systems registered and consuming OpenShift Container Platform entitlements for OpenShift Container Platform errata notification emails to generate.

This section will continue to be updated over time to provide notes on enhancements and bug fixes for future asynchronous errata releases of OpenShift sandboxed containers 1.5.

2.5.1. RHEA-2023:7493 - OpenShift sandboxed containers 1.5.0 image release, bug fix, and enhancement advisory

Issued: 2023-11-27

OpenShift sandboxed containers release 1.5.0 is now available. This advisory contains an update for OpenShift sandboxed containers with enhancements and bug fixes.

The list of bug fixes included in the update is documented in the RHEA-2023:7493 advisory.

2.5.2. RHBA-2024:0147 - OpenShift sandboxed containers 1.5.1 image release and bug fix advisory

Issued: 2024-01-11

OpenShift sandboxed containers release 1.5.1 is now available. This advisory contains an update for OpenShift sandboxed containers with bug fixes.

The list of bug fixes included in the update is documented in the RHBA-2024:0147 advisory.

2.5.3. RHBA-2024:0815 - OpenShift sandboxed containers 1.5.2 image release and bug fix advisory

Issued: 2024-02-15

OpenShift sandboxed containers release 1.5.2 is now available. This advisory contains an update for OpenShift sandboxed containers with bug fixes.

The list of bug fixes included in the update is documented in the RHBA-2024:0815 advisory.