OpenShift Sandboxed Containers 1.4

OpenShift sandboxed containers release notes

For Red Hat OpenShift

Last Updated: 2023-07-27

OpenShift Sandboxed Containers 1.4 OpenShift sandboxed containers release notes

For Red Hat OpenShift

Legal Notice

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

CHAPTER 1. OPENSHIFT SANDBOXED CONTAINERS 1.4 RELEASE NOTES

1.1. ABOUT THIS RELEASE

These release notes track the development of OpenShift sandboxed containers 1.4 alongside Red Hat Red Hat OpenShift 4.13.

This product is fully supported and enabled by default as of Red Hat OpenShift 4.10.

1.2. NEW FEATURES AND ENHANCEMENTS

1.2.1. Peer pods support for OpenShift sandboxed containers (Technology Preview)

Users can now deploy OpenShift sandboxed containers workloads using peer pods on either AWS or Microsoft Azure. 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.

1.2.2. QEMU error log collection

QEMU warning and error logs now print to both the node journal, the Kata runtime logs, and the CRI-O logs. For more information, see Viewing debug logs for OpenShift sandboxed containers.

1.2.3. Updated channel for installing OpenShift sandboxed containers Operator

The subscription channel when installing OpenShift sandboxed containers Operator is now always **stable**, instead of **stable-<version>** to enable consistency.

1.3. BUG FIXES

- Previously, upgrading OpenShift sandboxed containers did not automatically update the existing **KataConfig** CR. As a result, monitor pods from previous deployments were not restarted and continued to run with an outdated **kataMonitor** image.
  Starting from release 1.3.2, the **kataMonitorImage** was removed from the **KataConfig** CR, and the upgrade for all monitor pods is handled internally by the Operator.

  (KATA-1650)

- Previously, users could not install OpenShift sandboxed containers on a disconnected cluster. The pull specification of the kata-monitor container image used a tag instead of a digest. This prevented the image from being mirrored with the **ImageContentSourcePolicy** resource. With this release, the CSV **spec.relatedImages** section has been updated to ensure that all container images in the OpenShift sandboxed containers Operator are included. As a result, all container pull specifications now utilize digests instead of tags, enabling the installation of OpenShift sandboxed containers in disconnected environments.

  (KATA-2038)

- Previously, metrics were not available for OpenShift sandboxed containers running on a tainted node. With this release, a toleration has been added to the **kata-monitor** pods, enabling the pods to run and collect metrics on any node, even a tainted node. (KATA-2121)

- Previously, the base images for the OpenShift sandboxed containers Operator used **ubi8**/**ubi-mimimal** images. With this release, to ensure compatibility with RHEL 9 clusters and Red Hat OpenShift 4.13, the base images have been updated to use **ubi9**/**ubi** images. (KATA-2212)

1.4. KNOWN ISSUES

- If you are using OpenShift sandboxed containers, you might receive SELinux denials when accessing files or directories mounted from the **hostPath** volume in an Red Hat OpenShift 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)

- The progress of the runtime installation is shown in the **status** section of the **kataConfig** custom resource (CR). However, the progress is not shown if all of the following conditions are true:

  - There are no worker nodes defined. You can run **oc get machineconfigpool** to check the number of worker nodes in the machine config pool.

- No **kataConfigPoolSelector** is specified to select nodes for installation.

In this case, the installation starts on the control plane nodes because the Operator assumes it is a converged cluster where nodes have both control plane and worker roles. The **status** section of the **kataConfig** CR is not updated during the installation. (KATA-1017)

- In the **KataConfig** tab in the web console, if you click **Create KataConfig** while in the **YAML view**, the **KataConfig** YAML is missing the **spec** fields. Toggling to the **Form view** and then back to the **YAML view** fixes this issue and displays the full YAML. (KATA-1372)

- In the **KataConfig** tab in the web console, a **404: Not found** error message appears whether a **KataConfig** CR already exists or not. To access an existing **KataConfig** CR, go to **Home** > **Search**. From the **Resources** list, select **KataConfig**. (KATA-1605)

- During the installation of the **KataConfig** CR, the node status will be incorrect if the **KataConfig** CR deletion is initiated before the first node reboots. When this happens, the Operator is stuck in a state where the Operator attempts to delete and install the **KataConfig** CR simultaneously. The expected behavior is that the installation stops and the **KataConfig** CR is deleted. (KATA-1851)

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

```
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)

- When stopping a sandboxed container workload, the following QEMU error messages are logged to the worker node system journal:

```
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
```

These errors are harmless and can be ignored.

For more information on how to access the system journal logs, see Collecting OpenShift sandboxed containers data for Red Hat Support.

(KATA-2133)

- When installing the OpenShift sandboxed containers Operator using the web console, the UI might display the incorrect operator version after clicking **Install**.
The incorrect version appears in gray text in the installation window and reads:

<Version number> provided by Red Hat

The correct operator is installed. You can navigate to **Operators → Installed Operators** to see the correct version listed beneath the OpenShift sandboxed containers Operator.

(KATA-2161)

- When using peer pods with OpenShift sandboxed containers, the **kata-remote-cc** runtime class is created when you create the **KataConfig** CR and set the **enablePeerPods** field to **true**. As a result, users should see the **kata-remote-cc** runtime class in the **KataConfig** CR, in addition to the **kata** runtime class, and users should technically be able to run both standard Kata pods and peer-pod Kata pods on the same cluster.
  As a cluster admin, when you examine the **KataConfig** CR, you will only find **kata** in the **Status.runtimeClass** field. The runtime class **kata-remote-cc** does not appear. Currently, there is no workaround for this issue.

  (KATA-2164)

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

1.5. LIMITATIONS

- When using older versions of the Buildah tool in OpenShift sandboxed containers, the build fails with the following error:

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

  subprocess exited with status 1
  ```

  You must use the latest version of Buildah, available at quay.io.

  (KATA-1278)

1.6. ASYNCHRONOUS ERRATA UPDATES

Security, bug fix, and enhancement updates for OpenShift sandboxed containers 4.13 are released as asynchronous errata through the Red Hat Network. All Red Hat OpenShift 4.13 errata are available on the Red Hat Customer Portal . For more information about asynchronous errata, see the Red Hat OpenShift Life Cycle.

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 by email whenever new errata relevant to their registered systems are released.

NOTE

Red Hat Customer Portal user accounts must have systems registered and consuming Red Hat OpenShift entitlements for Red Hat OpenShift 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 1.4.

1.6.1. RHBA-2023:3529 - OpenShift sandboxed containers 1.4.0 image release, bug fix, and enhancement advisory

Issued: 2023-06-08

OpenShift sandboxed containers release 1.4.0 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 RHBA-2023:3529 advisory.

1.6.2. RHSA-2023:4290 - OpenShift sandboxed containers 1.4.1 image release, bug fix, and security advisory

Issued: 2023-07-27

OpenShift sandboxed containers release 1.4.1 is now available. This advisory contains an update for OpenShift sandboxed containers with security and bug fixes.

The list of bug fixes included in the update is documented in the RHSA-2023:4290 advisory.