In StarlingX, kata container is supported in master branch since Jan 23 of 2020.

By default, all pods are created with the default runtime configured in containerd, which is runc. If you want to launch pod with kata, you need declare it explicitly.

There are two methods to run kata container in Kubernetes.

1. By runtime class.

You need create a RuntimeClass with “handler” set to “kata”. Then reference this class in the Pod spec. RuntimeClass is supported in Kubernetes since v1.12.0. And it is the suggested method to run kata container.

Here is an example:

kind: RuntimeClass

apiVersion: node.k8s.io/v1beta1

metadata:

 name: kata-containers

handler: kata

---

apiVersion: v1

kind: Pod

metadata:

 name: busybox-runtime

spec:

 runtimeClassName: kata-containers

 containers:

 - name: busybox

 command:

 - sleep

 - "3600"

image: busybox

1. By annotation.

If “io.kubernetes.cri.untrusted-workload” is set to “true” in annotations, the pod will be run with kata container. This method is deprecated and may not be supported with future Kubernetes release. 1st method is recommended.

Here is an example:

apiVersion: v1

kind: Pod

metadata:

 name: busybox-untrusted

 annotations:

 io.kubernetes.cri.untrusted-workload: "true"

spec:

 containers:

 - name: busybox

 command:

 - sleep

 - "3600"

 image: busybox

Containerd is configured to translate upper setting in /etc/containerd/config.toml.

For RuntimeClass, here is the configuration in config.toml:

 [plugins.cri.containerd.runtimes.kata]

 runtime\_type = "io.containerd.kata.v2"

For annotation, here is the configuration in config.toml:

 [plugins.cri.containerd.runtimes.untrusted]

 runtime\_type = "io.containerd.kata.v2"

kata.v2 means shimv2(containerd-shim-kata-v2), which help Kubernetes to launch Pod and OCI-compatible containers with one shim per Pod.

Here are 2 methods to check whether the pod is running with kata container or not:

1. By run “uname -a” in both container and host. Host kernel version should be 3.10.0, while container kernel version should be 4.19 or higher. For normal container, host kernel version is the same as container’s.
2. By run “ps aux” in host, normal container is triggered by containerd-shim-runc-v1, while kata container is triggered by containerd-shim-kata-v2.

Appendix:

1. In case you have interest in the implementation of this feature, here is the links:

Spec file: <https://opendev.org/starlingx/specs/src/branch/master/doc/source/specs/stx-3.0/approved/containerization-2006145-kata-containers-integration.rst>

Story: <https://storyboard.openstack.org/#!/story/2006145>

Patches: <https://review.opendev.org/#/q/topic:kata+projects:starlingx>

1. Kata container is supported for Kubernetes only, since Kubernetes is the only supported container orchestration tool in StarlingX. Kata container support in docker is not implemented. Docker runtime also may be removed future in StarlingX, since all containers in StarlingX are run by Kubernetes now.

In case you want to try kata container with docker by yourself in StarlingX, please refer below link:

<https://github.com/kata-containers/documentation/blob/master/install/docker/centos-docker-install.md>

1. To support kata container, CRI runtime in Kubernetes is switched from dockershim to containerd. So you cannot view/operate Kubernetes’ container with docker client. Please use “crictl” instead, which supports similar command as docker. There is no difference for kubectl command before and after the switch.
2. More info for kata/containerd/Kubernetes RuntimeClass:

<https://katacontainers.io/>

<https://containerd.io/>

<https://kubernetes.io/docs/concepts/containers/runtime-class/>