Kata-Container

背景

1. 解决的问题。 每个容器/pod单独一个kernel， 提供**VM级别的负载隔离和安全性**， 恶意代码无法再利用共享内核来访问邻近的容器
2. Kata containers其实跟RunC类似，也是一个符合OCI运行时规范的一种 实现，不同之处是，它给每个容器（在Docker容器的 角度）或每个Pod（k8s的角度）增加了一个独立的linux内核（不共享宿主机的内核），使容器有更好 的隔离性，安全性。
3. 前身。 runV 以及 intel 的 clear Container 项目

容器生态中的位置

容器运行时是一个相对的概念，比如，从k8s的角度看，直接创建容器的组件是docker或containerd， 因此，将docker、**containerd以及**新加入的**CRI-O**作为容器运行时组件。而在docker、containerd或 CRI-O的角度看，真正启动容器的组件是runC，因此，docker中将runC作为容器运行时工具，当然在 docker中，**runC可以被替换，比如可以替换为本文介绍的kata containers**（即clear Container或者 runV)(角度不同，对象不同)

核心特性

1. 安全性。每个容器/pod单独一个kernel， 提供VM级别的负载隔离和安全性
2. 兼容性。能够支持不同平台的硬件 （x86-64，arm等）； 符合OCI(Open Container Initiative)规范； 兼容k8s的 CRI（Container Runtime Interface）接口规范
3. 性能。 兼容虚拟机的安全和容器的轻量特点。
4. Kata Containers represents a Kubelet pod as a VM
5. 存储。 默认virtio-fs
6. 网络。 Support CNI

v2新增特性

https://github.com/kata-containers/kata-containers/releases/tag/2.0.0

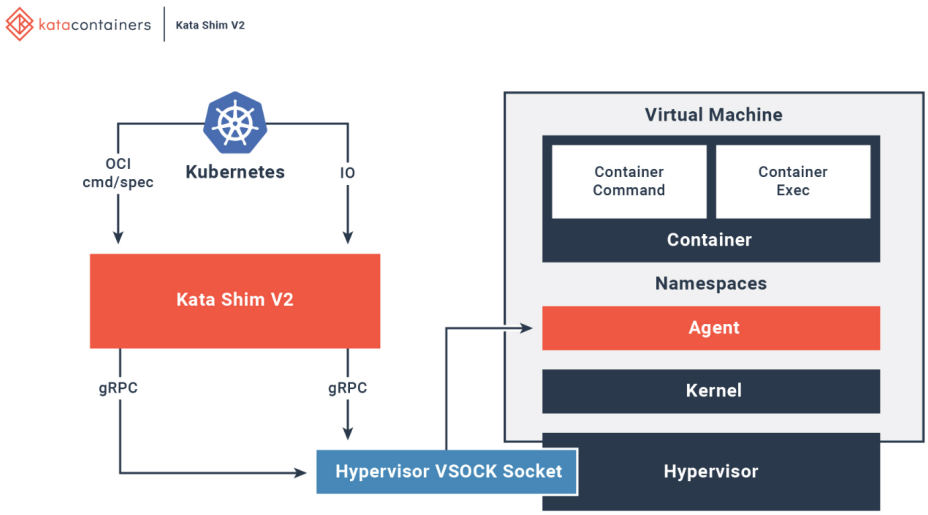
成为 Kata Containers 开发者 Day 2 - 人间指南

https://blog.csdn.net/yuchunyu97/article/details/109241723

1. Agent 用rust重写， 性能提升
2. agent通信协议改为ttrpc

3. virtio-9p改为Virtio-fs
4. QEMU的升级
5. shimv2组件只剩shimV2和agent
6. 基于promethus更完善的监控指标

架构



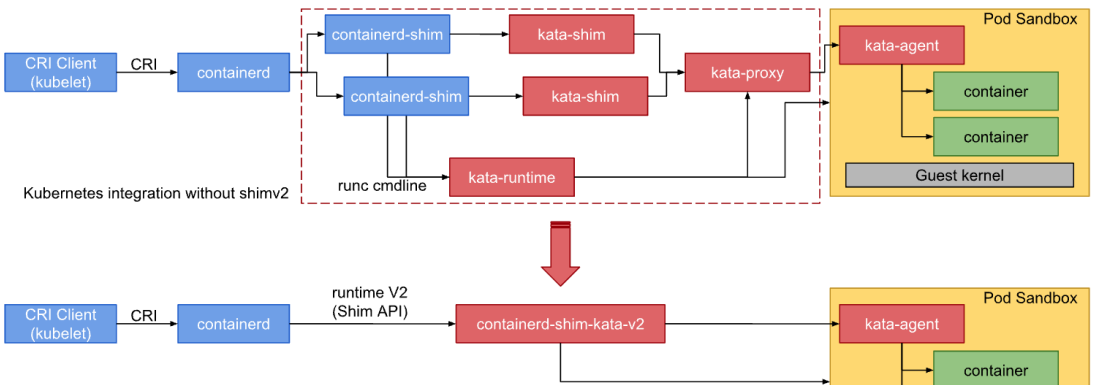
流程

https://github.com/kata-containers/kata-containers/tree/main/docs/design/architecture

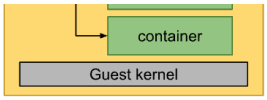
1. Kubelet -> containerd->(containerd-shim->kata-shim/kata-runtime->kata-proxy)->(kata-agent->container)

核心组件

最新的版本中， containerd-shim-kata-v2 集成了runtime+proxy+shim

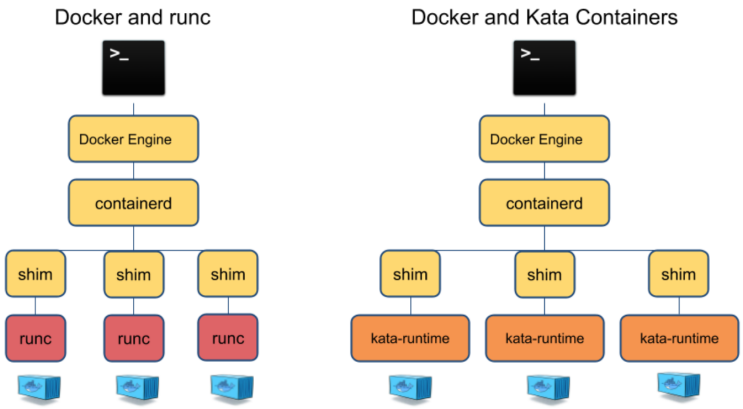


Kubernetes integration with shimv2



Runtime

1. Kata Containers runtime (kata-runtime)通过QEMU*/KVM技术创建了一种轻量型的虚拟机，兼容 OCI runtime specification 标准，支持Kubernetes* Container Runtime Interface (CRI)接口，可替换CRI shim runtime (runc) 通过k8s来创建pod或容器。
2. 符合 OCI 规范的容器运行时工具。主要用来创建轻量级虚拟机并通过 Agent 控制虚拟机内容器的生命周期。



Agent

1. rust实现
2. daemon进程（Pod sandbox内）
3. grpc服务。 代理容器和kata-runtime(proxy)的通信交互(ttrpc)
4. 在虚机内管理容器的生命周期

Proxy

1. Kata-runtime和agent的通信代理

Shim

1. Shim 相当于 Containerd-Shim 的适配，用来**处理容器进程的 stdio 和 signals**。Shim 可以将 Containerd-Shim 发来的数据流传给 Proxy，Proxy 再将数据流传输给微型虚拟机中的 Agent，Agent 传输给容器并执行相应的动作，同时 Shim 也支持将内部 Agent 产生的信号传输给 Proxy，Proxy 再传输给 Shim。

Kernel

1. QEMU/KVM虚拟机

原生kata缺点

https://www.modb.pro/db/156512

主要包括启动速度、资源消耗（cpu/mem）、稳定性

安装部署

依赖

1. K8s环境。 CRI为 containerd or CRI-O CRI-shims
2. nested virtualization or bare metal

kata安装

官方安装指南

1. 通过Damonset方式部署， 有k3s/rancher等限制
2. kata命令行工具 recomand

Shell

```
$ sudo -E dnf install -y centos-release-advanced-virtualization
$ sudo -E dnf module disable -y virt:rhel
$ source/etc/os-release
$ cat <<EOF| sudo -E tee /etc/yum.repos.d/kata-containers.repo  [kata-containers]  name=Kata Containers  baseurl=http://mirror.centos.org/\$contentdir/\$releasever/virt/\$basearch/kata-containers  enabled=1  gpgcheck=1  skip_if_unavailable=1  EOF$ sudo -E dnf install -y kata-containers
```

CRI-O配置

1. [以CRI-O为例](https://github.com/kata-containers/kata-containers/blob/main/docs/how-to/run-kata-with-k8s.md#cri-o)。 Containerd 参考 https://github.com/kata-containers/kata-

containers/blob/main/docs/how-to/how-to-use-k8s-with-cri-containerd-and-kata.md

2. CRI-O安装。 https://github.com/cri-o/cri-o/blob/main/tutorial.md
3. `An equivalent shim implementation for CRI-O is planned`. **CRI-O 还不支持 shimv2?**

配置文件

1. 默认路径 /etc/crio/crio.conf -> [crio.runtime] (配置项)
2. 完整的配置说明 https://github.com/cri-o/cri-o/blob/main/docs/crio.conf.5.md
3. CRI-O配置修改完, 执行 `sudo systemctl restart crio`， 使配置生效

Runtime Class

1. Add as runtime handler。 在/etc/crio/crio.conf.d， 新增如下子配置文件

C#

```
[crio.runtime.runtimes.kata]
    runtime_path = "/usr/bin/containerd-shim-kata-v2"
    runtime_type = "vm"
    runtime_root = "/run/vc"
    privileged_without_host_devices = true
```

网络 & 存储

验证

1. Runtime Class

Shell

```
$ cat > runtime.yaml <<EOF
apiVersion: node.k8s.io/v1
kind: RuntimeClass
metadata:
  name: kata
handler: kata
EOF

$ sudo -E kubectl apply -f runtime.yaml
```

2. Pod指定RC

Bash

```
$ cat << EOF | tee nginx-kata.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-kata
spec:
  runtimeClassName: kata
  containers:
  - name: nginx
    image: nginx

EOF
```

3. Create the pod。 `sudo -E kubectl apply -f nginx-kata.yaml`
4. 检查是否正常运行

Bash

```
// pod

$ sudo -E kubectl get pods
// Check hypervisor is running
$ ps aux | grep qemu
```

对接验证

1.11

1. 安装。 http://download.opensuse.org/repositories/home:/katacontainers:/releases:/x86_64:/stable-1.11/CentOS_7/ ； https://timchenxiaoyu.github.io/container/katacontainers/install.html
2. Kata版本， 1.11.0； k8s版本： 1.16

```
[ecf@cdn-k8s-m154 ~]$ kata-runtime -v
kata-runtime  : 1.11.0-rc0
   commit   : f7f5d42390b15b416f198570d7778dc09725a1d0
   OCI specs: 1.0.1-dev
```

3. cri-o配置。 runtime_type?

```
# Kata Containers with the default configured VMM
[crio.runtime.runtimes.kata-runtime]
runtime_path = "/bin/kata-runtime"
runtime_type = "oci"
```

```
runtime_root = ""

# Kata Containers with the QEMU VMM
#[crio.runtime.runtimes.kata-qemu]

# Kata Containers with the Firecracker VMM
#[crio.runtime.runtimes.kata-fc]
```

4. Runtimeclass

```
[root@cdn-k8s-m154 kata]# kubectl get runtimeclass -o yaml
apiVersion: v1
items:
- apiVersion: node.k8s.io/v1beta1
  handler: kata-runtime
  kind: RuntimeClass
  metadata:
    creationTimestamp: "2020-06-09T10:30:03Z"
    name: kata-runtime
    resourceVersion: "2742231"
    selfLink: /apis/node.k8s.io/v1beta1/runtimeclasses/kata-runtime
    uid: 90617653-e636-431e-8b6c-e2419e911159
kind: List
metadata:
  resourceVersion: ""
  selfLink: ""
```

5. Pvc

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name:  pvc-kata-test2
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Mi
  storageClassName: rbd-evm
```

6. 试验pod

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-kata1
spec:
  runtimeClassName: kata-runtime
  containers:
  - name: nginx
    image: nginx
    volumeMounts:
        - mountPath: /usr/share/nginx/html
          name: wwwroot
  volumes:
      - name: wwwroot
        persistentVolumeClaim:
          claimName:  pvc-kata-test2
```

7. 试验结果

CSS

```
// 主机的内核版本
Linux cdn-k8s-m154 5.5.7-1.el7.elrepo.x86_64 #1 SMP Fri Feb 28 12:21:58 EST 2020 x86_64 x86_64 x86_64 GNU/Linux

// kata容器的内核版本
Linux nginx-kata1 5.4.32-62.2.container #1 SMP Thu Jan 1 00:00:00 UTC 1970 x86_64 GNU/Linux

------------

// 挂载试验
[root@cdn-k8s-m154 pvc-327a8c30-5f97-47fe-9c85-5528b078099b]# pwd
/var/lib/kubelet/pods/a8cd2a1c-901d-4455-99dd-34ef4a9bb962/volumes/kubernetes.io~rbd/pvc-327a8c30-5f97-47fe-9c85-5528b078099b
[root@cdn-k8s-m154 pvc-327a8c30-5f97-47fe-9c85-5528b078099b]# ll
total 13
drwx------ 2 root root 12288 May 23 16:07 lost+found
-rw-r--r-- 1 root root    10 May 23 16:10 text.txt


```

2.x升级和验证

这里选用目前最新的稳定版本 2.4.1

1. 下载安装包，解压到/opt
2. 建立软链，拷贝默认的配置

Groovy

```
ln -s /opt/kata/bin/containerd-shim-kata-v2 /usr/local/bin/containerd-shim-kata-v2

ln -s /opt/kata/bin/kata-runtime /usr/local/bin/kata-runtime
mkdir -p /etc/kata-containers/

cp /opt/kata/share/defaults/kata-containers/configuration.toml /etc/kata-containers/
```

3. 修改CRIO的配置， 并重启 `sudo systemctl restart crio`

C#

```
// crio 配置
[crio.runtime.runtimes.kata-shimv2]
runtime_path = "/usr/local/bin/containerd-shim-kata-v2"
runtime_type = "vm"
manage_network_ns_lifecycle = true
```

4. 新建runtimeclass 指定handle为kata-shimv2

YAML

```
apiVersion: node.k8s.io/v1beta1
kind: RuntimeClass
metadata:
  name: kata-shimv2
handler: kata-shimv2
```

5. 新建验证nginx pod

YAML

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-kata-v2
spec:
  runtimeClassName: kata-shimv2
  containers:
  - name: nginx
    image: nginx
```

版本依赖

k8s: 1.16。 测试完发现k8s 1.16+ crio 1.18， kata 2.x报错

	crio-1.18.0
Kata 1.8	yes
Kata 2.0.1	No(required env variables [CNI_NETNS] missing)
Kata 2.1.0	No(required env variables [CNI_NETNS] missing)
Kata 2.3.0	

Kata 2.4.1	No(dial unix /run/containerd/s/607536384b30d59f92166babe47209b9f62b38169287d6dc6f5e17e579918771: connect: connection refused)

常用命令

1. journalctl -xe|grep kata
2. systemctl status crio -l
3. kubectl delete -f nginx-kata-v2.yaml --force --grace-period=0

性能优化

网络

存储

refer 资料

1. Github, https://github.com/kata-containers/kata-containers
2. 官方文档。 https://katacontainers.io/docs/
3. 整体介绍。 https://www.huweihuang.com/kubernetes-notes/runtime/kata/kata-container.html done
    a. https://mp.weixin.qq.com/s/YeMSdz9f1YVTEhQYFEFIAw done
    b. https://www.cnblogs.com/xiaochina/p/12812158.html done
4. 实践。
    a. https://www.infoq.cn/video/Tpf08PLZz8UgyF8Q5Kf0
    b. https://xie.infoq.cn/article/8a5dfbaa9a900d8e3e206bb60 done
5. k8s x kata相关官方文档
    a. with containerd
    b. kata对k8s的支持
    c. cri对接kata的配置