CNUTCon 2017 全球运维技术大会

Docker系统容器实践

黄强 / h.huangqiang@huawei.com
华为软件工程师

主办方 Geekbang 极客邦科技 InfoQ

Geekbang.
极客邦科技

极客搜索

全站干货，一键触达，只为技术

s.geekbang.org

扫描二维码立即体验

有没有一种搜索方式，能整合 InfoQ 中文站、极客邦科技旗下12大微信公众号矩阵的全部资源?
极客搜索，这款针对极客邦科技全站内容资源的轻量级搜索引擎，做到了!

扫描上方二维码，极客搜索!

EGO EXTRA GEEKS' ORGANIZATION NETWORKS

这里只有
技术领导者

EGO会员第二季招募季正式开启

报名时间：9月1日-9月15日

扫描添加E小欧，

邀您进入EGO会员预报名群

立即报名

E小欧

TABLE OF **CONTENTS**

什么是系统容器

- Pid 1进程通常是/sbin/init
- 容器内通常有一些系统服务
- 像使用虚拟机一样使用容器

系统容器 VS 应用容器

	系统容器	应用容器
容器性能	高	高
启动时间	慢	快
资源损耗	较低	低
镜像大小	较大	小
应用适配	高	低

为什么需要系统容器

- From VM to container
  - Legacy application
  - Legacy application
  - Legacy application

TABLE OF **CONTENTS**

--system-container

- Dockerfile
- Seccomp
- Capabilities
- Env
- Oci-systemd-hook/other hooks
- Reboot

Oci-systemd-hook

- https://github.com/projectatomic/oci-systemd-hook
- OCI systemd hook enables users to run systemd in docker and OCI compatible runtimes such as runc without requiring --privileged flag.
- it does the following:
  - Mounts a tmpfs on /run and /tmp
  - If there is content in the container image's /run and /tmp that content will be copied onto the tmpfs.
  - Creates a /etc/machine-id based on the the containers UUID
  - Mounts the hosts /sys/fs/cgroups file systemd read-only into the container
  - /sys/fs/cgroup/systemd will be mounted read/write into the container.

TABLE OF **CONTENTS**

资源信息隔离

- 容器内的资源信息隔离
  - 容器内看到自己的/proc/meminfo, /proc/stat等
  - 容器内可以使用top命令
- 解决方案
  - lxcfs

容器内使用cgroup

- 容器内使用cgroup
  - 不能修改容器自身的cgroup配置
  - 容器内可以创建和管理新的cgroup

- 解决方案
  - Kernel：Cgroup namespace
  - Docker run --cgroup private | host | “” | container:id

- 限制
  - 需要--privileged

动态添加设备

- https://github.com/moby/moby/pull/8348
- 实现：
  - Nsenter – mknod
  - Modify cgroup
  - “Docker update –add/del-device” or “docker-tools”

动态挂卷（1）

- http://jpetazzo.github.io/2015/01/13/docker-mount-dynamic-volumes/
- 实现
  - Find filesystem
  - Find the device
  - Nsenter – mknod – mount to tmp – bind mount – unmounts tmp

动态挂卷（2）

- 限制
  - 只能针对块设备上的目录
  - 存在安全隐患

SELinux（1）

- 容器内使用SELinux
  - 容器内支持SELinux的所有操作
  - 使用容器rootfs自己的SELinux规则
- 难点
  - SELinux没有隔离

SELinux（2）

- 实现
  - 隔离容器和host的SELinux开关
    （https://chromium-review.googlesource.com/c/361464/）
  - 容器内挂载SELinuxfs
  - Relabel容器rootfs
  - 修改docker二进制的安全上下文

SELinux（3）

- 问题和限制
  - SELinux依然没有隔离
    - host和container共用同一套规则，如果rootfs不兼容，需要在host上关闭SELinux
    - 容器间共用同一套规则
  - 修改了host上的docker二进制安全上下文，影响host上的docker SELinux policy

容器内使用Systemd（1）

- 新版本systemd（>=231）会破坏容器的io
  - https://github.com/moby/moby/issues/27202
- 原因
  - Systemd close /dev/console breaks io copy

容器内使用Systemd（2）

- 解决（workaround）
  - 修改containerd处理io copy的方式，循环处理io.Copy
- 社区解决方案
  - 修改io处理方式，改用epoll来实现
  - https://github.com/containerd/console/pull/10
  - https://github.com/containerd/containerd/pull/1259

TABLE OF **CONTENTS**

容器内使用LVM

- LVM、fdisk等操作在docker容器中无法正常使用
  - 容器的/dev使用tmpfs而不是devtmpfs
  - 现代的devtmpfs（2.6.32之后）负责创建device node，而不是udev
- 解决方案
  - 容器中要运行udev相关服务（容器中的udev可以收到udev kernel events）
  - 需要定制化容器中的udev rule
- 终极方案
  - https://www.linuxplumbersconf.org/2014/ocw/system/presentations/2157/original/Dynamic%20Device%20Management-v3.pdf

容器内的特权操作

- 建议非--privileged模式
- 默认只使能支持系统容器的最小权能
- 如果容器内需要其他操作（比如mount），需要自己通过--cap-add和--security-opt来修改权限控制

插入内核模块

- 没有对内核隔离性上做更多增强，新插入的内核模块将依赖所属子系统的内核隔离情况
- 不建议在启动多容器的环境上在容器中运行依赖内核模块的应用

TABLE OF **CONTENTS**

总结

- 通过适配和增强，docker容器也可以支持系统容器，实现更广泛的应用适配
- 在一些特性上还是存在限制，无法完全实现使用虚拟机的效果

THANKS!

智 能 时 代 的 新 运 维

CNUTCon 2017