cloudflight

Inspecting and Manipulating Running Containers

If you ever wanted to do stuff in a running container where the image doesn't include the necessary binaries, or you don't want to blow up your image with debugging stuff but still want to be able to debug some things lateron when the container is running this talk might give you the right hints.

Cloud Native Computing Linz Meetup - June 25, 2024

Martin Strigl

cloudflight

Ehlo!

Martin Strigl

Master of disaster (internal IT + SRE)
Not afraid of touching php
Using linux container technics since 2010

Why this talk ?

- I was asked if I can talk about something "not so everyday business" like
- Since I had the pleasure to deal with ugly situations in the past where manipulating running containers (especially in k8s context) was kind of necessary I thought this might be of benefit for others too

Picture Source: https://commons.wikimedia.org/wiki/File:Damaged_container_-_Port_of_Bremerhaven_-_2011.png

cloudflight

Topics covered

01

Local Manipulation

02

K8S/OKD: Manipulating Ressources

03

K8S/OKD: "Native" Tooling

04

K8S/OKD: "Native" Tooling extended

05

Final Words -- Q&A

Local Manipulation

- **Based on podman**
  - podman mount/unmount
    - Un/Mount a working container's root filesystem
  - podman un/pause
    - Un/Pause the processes in one or more containers
  - podman update
    - --blkio-weight: Block IO weight (relative weight) accepts a weight value between 10 and 1000
    - --cpus: Number of CPUs. The default is 0.000 which means no limit
    - --device-[read|write]-bps: Limit read/write rate (bytes per second) from a device
    - --device-[read|write]-iops: Limit read rate (IO per second) from a device
    - --memory: Memory limit
- **Based on nsenter**
  - --target <pid> target process to get namespaces from
  - --mount[=<file>] enter mount namespace
  - --net[=<file>] enter network namespace
  - --pid[=<file>] enter pid namespace

cloudflight

Topics covered

01

Local Manipulation

02

K8S/OKD: Manipulating Ressources

03

K8S/OKD: "Native" Tooling

04

K8S/OKD: "Native" Tooling extended

05

Final Words -- Q&A

K8S/OKD: Manipulating Ressources

- **Storage**
  - Cannot be changed during runtime - at least not the "clean" way
  - Depending on your storage provider you might have to do a lot of steps
  - Requires anyway high system privileges
  - Example steps for openstack cinder volumes which are based on ceph rbd and attached to openstack (k)vm's
    - On node "connected" to openstack api
      - OS_VOLUME_API_VERSION=3.42 cinder extend <volume-id> 100
    - On hypervisor which hosts OKD vm
      - virsh domblklist instance-00002016
      - virsh domblkinfo instance-00002016 sdb
      - virsh blockresize --domain instance-00002016 --path sdb --size 3000GiB
    - On OKD vm
      - echo 1 > /sys/class/scsi_device/3:0:0:4/device/rescan
      - xfs_growfs /var/lib/origin/openshift.local.volumes/pods/<<UUID>>/volumes/kubernetes.io~iscsi/pvc-<<UUID>>
- **CPU**
  - docker/podman/crictl update --cpus 4 CONTAINER-ID (syntax/cliswitchname differs slightly between runtimes)
- **Memory**
  - docker/podman/crictl update --memory-swap 5500M --memory 5500M CONTAINER-ID (syntax/cliswitchname differs slightly between runtimes)

cloudflight

Topics covered

01

Local Manipulation

02

K8S/OKD: Manipulating Ressources

03

K8S/OKD: "Native" Tooling

04

K8S/OKD: "Native" Tooling extended

05

Final Words -- Q&A

K8S/OKD: "Native" Tooling

- **kubectl debug**
  - See https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.25/#ephemeralcontainer-v1-core
  - e.g.: kubectl debug db-7bf8f455d5-27krw --target=mariadb --image alpine –it
  - Caveat:
    - if used on openshift pods, creates ephemeralcontainer tree in pod object which lateron prohibits usage of oc debug
    - "fills" up the ephemeral container tree in pod object -- cannot be deleted
  - Creates an additional (ephemeral-)container in the referenced pod
    - Has same network namespace
    - Has NOT same mounts
    - Has same pid namespace
    - Cannot have/define
      - Livenessprobe / readinessprobe
      - Ports
      - Ressources
      - Userid we should run as
- **oc debug**
  - oc debug --image=alpine -c mariadb --keep-init-containers=false --one-container --as-root db-6fcfc775bb-glbs5
  - Creates an additional pod which shares relevant namespaces
    - Has DIFFERENT network and pid namespace
    - Has same mounts

cloudflight

Topics covered

01

Local Manipulation

02

K8S/OKD: Manipulating Ressources

03

K8S/OKD: "Native" Tooling

04

K8S/OKD: "Native" Tooling extended

05

Final Words -- Q&A

K8S/OKD: "Native" Tooling extended

- **Adressing issues from oc/kubectl debug**
  - based on https://github.com/JonMerlevede/kubectl-superdebug => https://github.com/strima/kubectl-superdebug
    - Creates also ephemeralcontainers (pay attention)
    - Has same mounts
    - Has same network namespace
    - Has same pid namespace
    - Has same uid namespace
    - Default runs with same securitycontext as target container
      - Can be switched to root

cloudflight

Topics covered

01

Local Manipulation

02

K8S/OKD: Manipulating Ressources

03

K8S/OKD: "Native" Tooling

04

K8S/OKD: "Native" Tooling extended

05

Final Words -- Q&A

cloudflight

Thank you for your attention