Docker, Kubernetes, VMware Tanzu and RedHat OCP
07Oct2022
Day03
Docker Port-Forwarding, Docker Storage
----
what is container?
A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing.
what is runtime?
A container runtime, also known as container engine, is a software component that can run containers on a host operating system.
what is image
A container image is an unchangeable, static file that includes executable code so it can run an isolated process
what is registry
A container registry is a repository—or collection of repositories—used to store and access container images.
registry comes in to:
-public, accessible by public https://hub.docker.com/ https://catalog.redhat.com/
-private, accessible by specific person/organization
registry address in Docker
```
# cat /etc/containers/registries.conf
unqualified-search-registries = ["registry.fedoraproject.org", "registry.access.redhat.com", "registry.centos.org", "docker.io"]
```
pull image from specific registry
https://catalog.redhat.com/software/containers/rhel8/httpd-24/5ba0addbbed8bd6ee819856a?container-tabs=gti
```
# docker login registry.redhat.io
# docker pull registry.redhat.io/rhel8/httpd-24
```
Run container with delay syncing with Container-Host
```
# docker run -d --name apache --restart unless-stopped httpd
```
send/receive file to and from container
send
```
# touch /tmp/a.txt
# docker ps
8f518679af10   httpd    "httpd-foreground"   6 minutes ago   Up 6 minutes   80/tcp    apache
# docker cp /tmp/a.txt 8f518679af10:/tmp
# docker exec -it apache ls /tmp
a.txt
```
receive
```
# docker cp apache:/tmp/a.txt /var/tmp/
# ls /var/tmp/
a.txt
```

Docker Port Forwarding
access container from outside
-to get access container from outside need to use port-forwarding
-need to map port from Container-host on Container
```
# docker ps
8f518679af10   httpd    "httpd-foreground"   16 minutes ago   Up 16 minutes   80/tcp    apache
# docker exec -it apache curl localhost
OCI runtime exec failed: exec failed: container_linux.go:380: starting container process caused: exec: "curl": executable file not found in $PATH: unknown
# docker exec -it apache /bin/bash
root@8f518679af10:/usr/local/apache2# cat /etc/os-release
root@8f518679af10:/usr/local/apache2# apt-get update
root@8f518679af10:/usr/local/apache2# apt-get install curl
root@8f518679af10:/usr/local/apache2# curl localhost
<html><body><h1>It works!</h1></body></html>
root@8f518679af10: exit
# docker exec -it apache curl localhost
<html><body><h1>It works!</h1></body></html>
# docker inspect httpd
"ExposedPorts": {
            "80/tcp": {}
# docker run -d --name apache -p 80:80 --restart always httpd
# docker port apache
80/tcp -> 0.0.0.0:80
80/tcp -> :::80
<Container-Host port>:<Container-port>
```
open web browser
http://192.168.56.99
It works!

Mohammad Ali Naghval
ali@prodevans.com

Container Storage

container storage is ephemeral, if container will crash/delete data won't accessible.
connect Container to external storage.
to mount Container-Storage to External-Storage/Container-Host need to mount it through --volume, -v

```
# mkdir /root/html
# echo "Hello!" >/root/html/index.html
# tree /root/
/root/
├── ~
├── 1
├── anaconda-ks.cfg
└── html
    └── index.html
# docker run -d --name apache -p 80:80 -v /root/html:/usr/local/apache2/htdocs:Z httpd
```

Z sync SELinux security context between host and container

```
# docker exec -it apache /bin/bash
root@387dad6afc9a:/usr/local/apache2# apt-get update ; apt-get install curl
root@387dad6afc9a:/usr/local/apache2# exit
exit
# curl localhost
Hello!
```

open web browser
http://192.168.56.99
Hello!

```
# docker container rm apache -f
apache
# cat /root/html/index.html
Hello!
```

But if any issue happens on Container-Host, data won't accessible. solution: use File-Based or Block-Based protocol and send data out of cluster.

```
hostname: docker.example.com      ->Container-Host   ->192.168.56.99/24
hostname: generic.example.com     ->NFS-Sevrer       ->192.168.56.100/24
```

need to do basic configuration for both hosts:
1-hostname
2-network ip/mask, dns and gateway
3-repository

NFS-SERVER(External Storae)
-install nfs package
-enable/start it
-config firewalld

```
# yum install nfs-utils rpcbind -y
# systemctl enable nfs-server.service
# systemctl start nfs-server.service
# systemctl status nfs-server.service
# systemctl status firewalld.service
● firewalld.service
  Loaded: masked
# sestatus
SELinux status:              disabled
# lsblk
# fdisk /dev/sdb
n
e
press Enter
press Enter
press Enter
p
/dev/sdb1      2048 20971519 20969472  10G  5 Extended
n
Adding logical partition 5
press Enter
+1G
p
/dev/sdb1      2048 20971519 20969472  10G  5 Extended
/dev/sdb5      4096  2101247  2097152    1G 83 Linux
t
Partition number (1,5, default 5): press Enter
8e
p
/dev/sdb1      2048 20971519 20969472  10G  5 Extended
/dev/sdb5      4096  2101247  2097152   1G 8e Linux LVM
w
# udevadm settle
# lsblk
# fdisk -l /dev/sdb
# pvcreate /dev/sdb5
# vgcreate vg1 /dev/sdb5
# lvcreate -n lv1 -l 100%FREE vg1
# lvs
 lv1  vg1  -wi-a----- 1020.00m
# mkfs.xfs /dev/mapper/vg1-lv1
# blkid
/dev/mapper/vg1-lv1: UUID="34df4e1f-34fa-4955-94af-c79c6357f189" BLOCK_SIZE="512" TYPE="xfs"
# mkdir /mnt/disk1
# ls -ld /mnt/disk1/
drwxr-xr-x 2 root root 6 Oct  7 21:10 /mnt/disk1/
# chmod 757 /mnt/disk1/
# ls -ld /mnt/disk1/
drwxr-xrwx 2 root root 6 Oct  7 21:10 /mnt/disk1/
# echo "/dev/mapper/vg1-lv1 /mnt/disk1/ xfs defaults 0 0" >>/etc/fstab
# mount -a
# df -hT
/dev/mapper/vg1-lv1   xfs      1014M   40M  975M   4% /mnt/disk1
# vim /etc/exports
/mnt/disk1 192.168.56.0/24(rw,sync)
/mnt/disk1 192.168.29.0/24(rw,sync)
:wq!
# systemctl restart nfs-server.service
# exportfs -rva
exporting 192.168.56.0/24:/mnt/disk1
exporting 192.168.29.0/24:/mnt/disk1
# showmount -e
Export list for generic.example.com:
/mnt/disk1 192.168.29.0/24,192.168.56.0/24
```

now, back to **Docker-Host(NFS-Client)**

```
# yum install nfs-utils -y
# showmount -e 192.168.29.233
Export list for 192.168.29.233:
/mnt/disk1 192.168.29.0/24,192.168.56.0/24
# mkdir /mnt/nfs
# mount -t nfs -o rw,sync 192.168.29.233:/mnt/disk1 /mnt/nfs
# df -hT
192.168.29.233:/mnt/disk1 nfs4     1014M   40M  975M   4% /mnt/nfs
```

NOTE: if won’t work try to check network connectivity and use ping command.

```
# docker search http
# docker pull httpd
# docker images
# docker inspect httpd
"ExposedPorts": {
            "80/tcp": {}
# mkdir /mnt/nfs/html
# echo "Hello!" >/mnt/nfs/html/index.html
# docker run -d --name apache -p 80:80 -v /mnt/nfs/html:/usr/local/apache2/htdocs:Z httpd
```

verify

```
# curl localhost
Hello!
```

open web browser

http://192.168.56.99

Hello!

back to NFS-Server and check

```
# ls /mnt/disk1/html/
index.html
# cat /mnt/disk1/html/index.html
Hello!
```