Lotus-容器运行-2k

2022年3月1日　　10:43

os使用ubuntu20.04.03server版本

整套使用rancher+k3s实现集群管理，对接私有harbor镜像仓库，后期对接运维审计，可对接jumpserver开源堡垒机使用

环境搭建不描述，主要描述dockerfile文件和编译注意事项以及在宿主机上执行lotus命令

1，编译前一定要使用git clone官方github的代码，不要下载官方指定版本的源码包，源码包缺少.git目录

为保障宿主机开执行lotus命令，首先要在宿主机上安装相关的依赖，依赖参照官方文档安装

```
sudo apt install mesa-opencl-icd ocl-icd-opencl-dev gcc git bzr jq pkg-config curl clang build-essential hwloc libhwloc-dev wget nfs-common -y
```

2，编辑dockerfile文件，源文件为官方dockerfile，根据自身需求自行修改

```
FROM golang:1.16.4 AS builder-deps
MAINTAINER Lotus Development Team
RUN echo 'deb http://mirrors.ustc.edu.cn/debian/ buster main contrib non-free\n\
deb http://mirrors.ustc.edu.cn/debian/ buster-updates main contrib non-free\n\
deb http://mirrors.ustc.edu.cn/debian-security/ buster/updates main contrib non-free\n'\
 >/etc/apt/sources.list
#golang镜像基于debian系统，添加上一个RUN，用于修改debian的源
RUN apt-get update && apt-get install -y ca-certificates build-essential clang ocl-icd-opencl-dev ocl-icd-libopencl1 jq libhwloc-dev

ARG RUST_VERSION=nightly
ENV XDG_CACHE_HOME="/tmp"

ENV RUSTUP_HOME=/usr/local/rustup \
    CARGO_HOME=/usr/local/cargo \
    PATH=/usr/local/cargo/bin:$PATH

RUN wget "https://static.rust-lang.org/rustup/dist/x86_64-unknown-linux-gnu/rustup-init"; \
    chmod +x rustup-init; \
    ./rustup-init -y --no-modify-path --profile minimal --default-toolchain $RUST_VERSION; \
    rm rustup-init; \
    chmod -R a+w $RUSTUP_HOME $CARGO_HOME; \
    rustup --version; \
    cargo --version; \
    rustc --version;


FROM builder-deps AS builder-local
MAINTAINER Lotus Development Team

COPY ./ /opt/filecoin
WORKDIR /opt/filecoin
ENV https_proxy http://192.168.1.10:1080
ENV http_proxy http://192.168.1.10:1080
ENV GOPROXY https://goproxy.cn
#添加上面三个ENV变量，用于代理go和https和http，主要用于通过代理下载github上的源码和go的相关依赖
RUN make clean deps


FROM builder-local AS builder
MAINTAINER Lotus Development Team

WORKDIR /opt/filecoin

ARG RUSTFLAGS=""
ARG GOFLAGS=""

RUN make lotus lotus-miner lotus-worker lotus-shed lotus-wallet lotus-gateway lotus-stats
#上述为编译所有lotus程序，可以根据自身需求选择性编译，如果要变异2k网，则使用make debug，注意编译2k网时候添加lotus-stats选项如下所示
#RUN make debug lotus-stats

FROM ubuntu:20.04 AS base
MAINTAINER Lotus Development Team

# Base resources
COPY --from=builder /etc/ssl/certs                        /etc/ssl/certs
COPY --from=builder /lib/x86_64-linux-gnu/libdl.so.2        /lib/
COPY --from=builder /lib/x86_64-linux-gnu/librt.so.1        /lib/
COPY --from=builder /lib/x86_64-linux-gnu/libgcc_s.so.1      /lib/
COPY --from=builder /lib/x86_64-linux-gnu/libutil.so.1      /lib/
COPY --from=builder /usr/lib/x86_64-linux-gnu/libltdl.so.7   /lib/
COPY --from=builder /usr/lib/x86_64-linux-gnu/libnuma.so.1   /lib/
COPY --from=builder /usr/lib/x86_64-linux-gnu/libhwloc.so.5  /lib/
COPY --from=builder /usr/lib/x86_64-linux-gnu/libOpenCL.so.1 /lib/

RUN useradd -r -u 532 -U fc
#上述添加用户fc，可以自行修改，修改后要同步到下面的配置中，同时检查makefile中是否有相关的用户名和uid

###
FROM base AS lotus
MAINTAINER Lotus Development Team
```

```
COPY --from=builder /opt/filecoin/lotus /usr/local/bin/
COPY --from=builder /opt/filecoin/lotus-shed /usr/local/bin/
COPY scripts/docker-lotus-entrypoint.sh /

ENV FILECOIN_PARAMETER_CACHE /var/tmp/filecoin-proof-parameters
ENV LOTUS_PATH /var/lib/lotus
ENV DOCKER_LOTUS_IMPORT_SNAPSHOT https://fil-chain-snapshots-fallback.s3.amazonaws.com/mainnet/minimal_finality_stateroots_latest.car
ENV DOCKER_LOTUS_IMPORT_WALLET ""

RUN mkdir /var/lib/lotus /var/tmp/filecoin-proof-parameters
RUN chown fc: /var/lib/lotus /var/tmp/filecoin-proof-parameters

VOLUME /var/lib/lotus
VOLUME /var/tmp/filecoin-proof-parameters

USER fc

EXPOSE 1234

ENTRYPOINT ["/docker-lotus-entrypoint.sh"]

CMD ["-help"]

###
FROM base AS lotus-wallet
MAINTAINER Lotus Development Team

COPY --from=builder /opt/filecoin/lotus-wallet /usr/local/bin/

ENV WALLET_PATH /var/lib/lotus-wallet

RUN mkdir /var/lib/lotus-wallet
RUN chown fc: /var/lib/lotus-wallet

VOLUME /var/lib/lotus-wallet

USER fc

EXPOSE 1777

ENTRYPOINT ["/usr/local/bin/lotus-wallet"]

CMD ["-help"]

###
FROM base AS lotus-gateway
MAINTAINER Lotus Development Team

COPY --from=builder /opt/filecoin/lotus-gateway /usr/local/bin/

USER fc

EXPOSE 1234

ENTRYPOINT ["/usr/local/bin/lotus-gateway"]

CMD ["-help"]


###
FROM base AS lotus-miner
MAINTAINER Lotus Development Team

COPY --from=builder /opt/filecoin/lotus-miner /usr/local/bin/
COPY scripts/docker-lotus-miner-entrypoint.sh /

ENV FILECOIN_PARAMETER_CACHE /var/tmp/filecoin-proof-parameters
ENV LOTUS_MINER_PATH /var/lib/lotus-miner

RUN mkdir /var/lib/lotus-miner /var/tmp/filecoin-proof-parameters
RUN chown fc: /var/lib/lotus-miner /var/tmp/filecoin-proof-parameters

VOLUME /var/lib/lotus-miner
VOLUME /var/tmp/filecoin-proof-parameters

USER fc

EXPOSE 2345

ENTRYPOINT ["/docker-lotus-miner-entrypoint.sh"]

CMD ["-help"]


###
FROM base AS lotus-worker
MAINTAINER Lotus Development Team

COPY --from=builder /opt/filecoin/lotus-worker /usr/local/bin/
```

```
ENV FILECOIN_PARAMETER_CACHE /var/tmp/filecoin-proof-parameters
ENV LOTUS_WORKER_PATH /var/lib/lotus-worker

RUN mkdir /var/lib/lotus-worker
RUN chown fc: /var/lib/lotus-worker

VOLUME /var/lib/lotus-worker

USER fc

EXPOSE 3456

ENTRYPOINT ["/usr/local/bin/lotus-worker"]

CMD ["-help"]


###
from base as lotus-all-in-one

ENV FILECOIN_PARAMETER_CACHE /var/tmp/filecoin-proof-parameters
ENV LOTUS_MINER_PATH /var/lib/lotus-miner
ENV LOTUS_PATH /var/lib/lotus
ENV LOTUS_WORKER_PATH /var/lib/lotus-worker
ENV WALLET_PATH /var/lib/lotus-wallet
ENV DOCKER_LOTUS_IMPORT_SNAPSHOT https://fil-chain-snapshots-fallback.s3.amazonaws.com/mainnet/minimal_finality_stateroots_latest.car

COPY --from=builder /opt/filecoin/lotus     /usr/local/bin/
COPY --from=builder /opt/filecoin/lotus-shed /usr/local/bin/
COPY --from=builder /opt/filecoin/lotus-wallet /usr/local/bin/
COPY --from=builder /opt/filecoin/lotus-gateway /usr/local/bin/
COPY --from=builder /opt/filecoin/lotus-miner /usr/local/bin/
COPY --from=builder /opt/filecoin/lotus-worker /usr/local/bin/
COPY --from=builder /opt/filecoin/lotus-stats /usr/local/bin/

RUN mkdir /var/tmp/filecoin-proof-parameters
RUN mkdir /var/lib/lotus
RUN mkdir /var/lib/lotus-miner
RUN mkdir /var/lib/lotus-worker
RUN mkdir /var/lib/lotus-wallet
RUN chown fc: /var/tmp/filecoin-proof-parameters
RUN chown fc: /var/lib/lotus
RUN chown fc: /var/lib/lotus-miner
RUN chown fc: /var/lib/lotus-worker
RUN chown fc: /var/lib/lotus-wallet


VOLUME /var/tmp/filecoin-proof-parameters
VOLUME /var/lib/lotus
VOLUME /var/lib/lotus-miner
VOLUME /var/lib/lotus-worker
VOLUME /var/lib/lotus-wallet

EXPOSE 1234
EXPOSE 2345
EXPOSE 3456
EXPOSE 1777
```

上述dockerfile修改完毕之后即可执行docker build命令，执行docker build命令要在git clone好的lotus目录下执行，传入的上下文为.

如果要是编译指定版本，则在编译前先通过git pull --all，然后执行 git checkout 版本号在执行docker build操作，例如编译1.14.2版本

```
git pull --all
git checkout v1.14.2
```

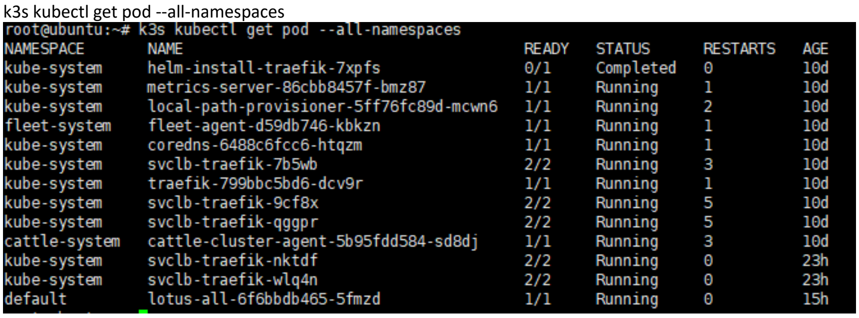
不指定版本默认则编译latest版本

编译完成后打包镜像，上传到harbor私有镜像库，对接rancher直接部署服务，部署之后，想要宿主机也可以执行lotus命令，将容器内（pod内）的程序拷贝至宿主机，即可在宿主机执行，由于拷贝之后的程序没有环境变量，想要成功执行lotus-miner info之类的命令则需要通过脚本实现，使用脚本增加相关的环境变量来实现

如下首先是拷贝主程序

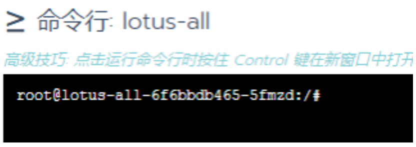
在k3s-master节点下执行，列出所有相关的pod以及关联的命名空间

k3s kubectl get pod --all-namespaces



```
root@ubuntu:~# k3s kubectl get pod --all-namespaces
NAMESPACE       NAME                                      READY   STATUS      RESTARTS   AGE
kube-system     helm-install-traefik-7xpfs                0/1     Completed   0          10d
kube-system     metrics-server-86cbb8457f-bmz87           1/1     Running     1          10d
kube-system     local-path-provisioner-5ff76fc89d-mcwn6   1/1     Running     2          10d
fleet-system    fleet-agent-d59db746-kbkzn                1/1     Running     1          10d
kube-system     coredns-6488c6fcc6-htqzm                  1/1     Running     1          10d
kube-system     svclb-traefik-7b5wb                       2/2     Running     3          10d
kube-system     traefik-799bbc5bd6-dcv9r                  1/1     Running     1          10d
kube-system     svclb-traefik-9cf8x                       2/2     Running     5          10d
kube-system     svclb-traefik-qggpr                       2/2     Running     5          10d
cattle-system   cattle-cluster-agent-5b95fdd584-sd8dj     1/1     Running     3          10d
kube-system     svclb-traefik-nktdf                       2/2     Running     0          23h
kube-system     svclb-traefik-wlq4n                       2/2     Running     0          23h
default         lotus-all-6f6bbdb465-5fmzd                1/1     Running     0          15h
```

通过进入rancher的pod的控制台，我们得知我们的pod名称为lotus-all-6f6bbdb465-5fmzd，如下图



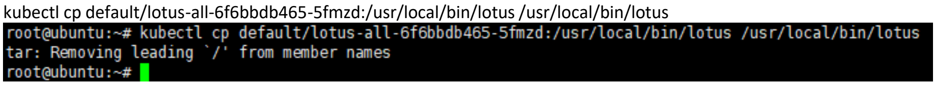
```
root@lotus-all-6f6bbdb465-5fmzd:/#
```

在宿主机上执行，拷贝文件到宿主机，在此案例是拷贝我们的lotus程序到宿主机，然后传送到运行该pod的主机上，通过脚本实现执行lotus命令

拷贝文件命令格式

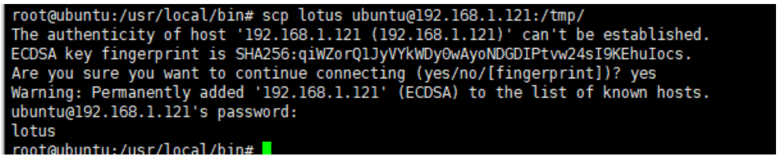
kubectl cp 命名空间/pod名称：文件路径 本地文件路径，例如拷贝pod内的lotus到本地的/usr/local/bin/目录下

kubectl cp default/lotus-all-6f6bbdb465-5fmzd:/usr/local/bin/lotus /usr/local/bin/lotus

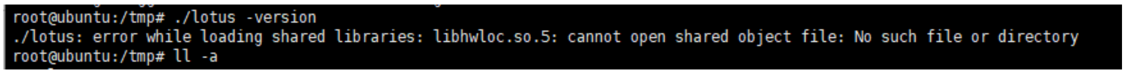


```
root@ubuntu:~# kubectl cp default/lotus-all-6f6bbdb465-5fmzd:/usr/local/bin/lotus /usr/local/bin/lotus
tar: Removing leading `/' from member names
root@ubuntu:~# 
```

然后在scp传送到运行该pod的主机上



```
root@ubuntu:/usr/local/bin# scp lotus ubuntu@192.168.1.121:/tmp/
The authenticity of host '192.168.1.121 (192.168.1.121)' can't be established.
ECDSA key fingerprint is SHA256:qiWZorQ1JyVYkWDy0wAyoNDGDIPtvw24sI9KEhuIocs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.121' (ECDSA) to the list of known hosts.
ubuntu@192.168.1.121's password:
lotus
root@ubuntu:/usr/local/bin# 
```



```
root@ubuntu:/tmp# ./lotus -version
./lotus: error while loading shared libraries: libhwloc.so.5: cannot open shared object file: No such file or directory
root@ubuntu:/tmp# ll -a
```

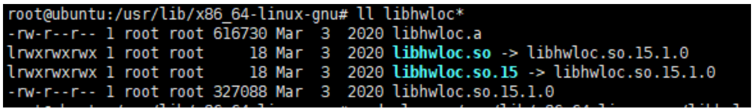
提示缺少共享库

安装好libhwloc-dev的话做个连接即可，

查看连接关系

cd /usr/lib/x86_64-linux-gnu

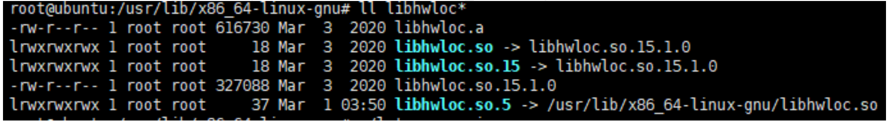
ll libhwloc*



```
root@ubuntu:/usr/lib/x86_64-linux-gnu# ll libhwloc*
-rw-r--r-- 1 root root 616730 Mar  3  2020 libhwloc.a
lrwxrwxrwx 1 root root     18 Mar  3  2020 libhwloc.so -> libhwloc.so.15.1.0
lrwxrwxrwx 1 root root     18 Mar  3  2020 libhwloc.so.15 -> libhwloc.so.15.1.0
-rw-r--r-- 1 root root 327088 Mar  3  2020 libhwloc.so.15.1.0
```

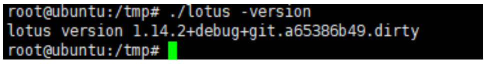
创建连接

sudo ln -s /usr/lib/x86_64-linux-gnu/libhwloc.so /usr/lib/x86_64-linux-gnu/libhwloc.so.5



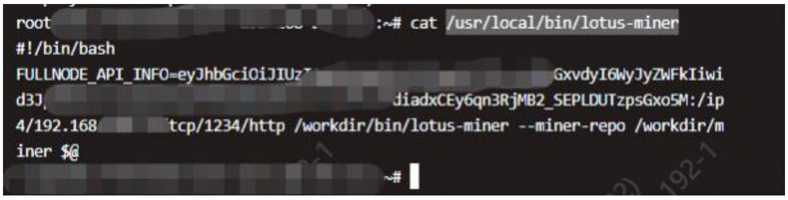
```
root@ubuntu:/usr/lib/x86_64-linux-gnu# ll libhwloc*
-rw-r--r-- 1 root root 616730 Mar  3  2020 libhwloc.a
lrwxrwxrwx 1 root root     18 Mar  3  2020 libhwloc.so -> libhwloc.so.15.1.0
lrwxrwxrwx 1 root root     18 Mar  3  2020 libhwloc.so.15 -> libhwloc.so.15.1.0
-rw-r--r-- 1 root root 327088 Mar  3  2020 libhwloc.so.15.1.0
lrwxrwxrwx 1 root root     37 Mar  1 03:50 libhwloc.so.5 -> /usr/lib/x86_64-linux-gnu/libhwloc.so
```

返回/tmp，尝试执行



```
root@ubuntu:/tmp# ./lotus -version
lotus version 1.14.2+debug+git.a65386b49.dirty
root@ubuntu:/tmp# 
```

如果需要执行更多命令，如果lotus-miner之类的，下面提供一种思路，通过拷贝pod内的主程序到指定主机，然后编辑运行脚本，调用拷贝过来的程序，脚本内添加一些运行所需的环境变量，例如下面的例子，是在lotus-miner主程序放在/workdir目录下，然后/usr/local/bin下的lotus-miner就是一个脚本，脚本的内容是调用/workdir下面的主程序，加上一些变量和运行参数一起执行。这样就可以实现在宿主机执行pod内的命令了。



```
root[redacted]:~# cat /usr/local/bin/lotus-miner
#!/bin/bash
FULLNODE_API_INFO=eyJhbGciOiJIUz[redacted]GxvdyI6WyJyZWFkIiwi
d3J[redacted]diadxCEy6qn3RjMB2_SEPLDUTzpsGxo5M:/ip
4/192.168[redacted]tcp/1234/http /workdir/bin/lotus-miner --miner-repo /workdir/m
iner $@
[redacted]~# 
```