环境构建

一、Docker 安装

```
 # 1、卸载docker
yum remove docker \
                  docker-client \
                  docker-client-latest \
                  docker-common \
                  docker-latest \
                  docker-latest-logrotate \
                  docker-logrotate \
                  docker-engine

# 2、需要的安装包
yum install -y yum-utils

# 3、设置镜像的仓库，默认是国外的
yum-config-manager \
    --add-repo \
    http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

# 更新软件包索引（可选）
yum makecache fast

# 4、安装 docker 相关内容（最新版） docker-ce(社区版)  ee (企业版)
yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin

# 5、启动 docker
systemctl start docker

# 6、验证 docker
docker version

# 7、docker 运行 hello-world
docker run hello-world

# 8、查看这个 hello-world 镜像
docker images

[root@VM-8-6-centos /]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    feb5d9fea6a5   9 months ago   13.3kB

```

```
# 9、卸载 Docker
# 删除依赖
yum remove docker-ce docker-ce-cli containerd.io docker-compose-plugin
# 删除资源
rm -rf /var/lib/docker
rm -rf /var/lib/containerd

# 配置镜像加速
vim /etc/docker/daemon.json
{
"registry-mirrors": ["https://mirror.ccs.tencentyun.com"]
}
systemctl restart docker

# 10、设置开机启动
systemctl enable docker.service
```

二、自定义网络

自定义网络好处：防止因docker重启引起容器ip变化。

此处有疑问：

1. 物理机ip是192.168.0.109
2. 创建的docker网络是192.168.0.0/16，导致VM不能ping通物理机，但是能连接外网
3. 后修改docker自定义网络为172.18.0.0/16，得以正常
4. **待思考，为啥？**

```
# 创建webnet网络
# docker network create --driver bridge --subnet 192.168.0.0/16 --gateway 192.168.0.1 webnet
docker network create --driver bridge --subnet 172.18.0.0/16 --gateway 172.18.0.1 webnet

# 查看所有网络
docker network ls

# 查看webnet网络信息
docker network inspect webnet
```

三、MySQL 安装

```
# 新建目录
mkdir -p /kwang/soft/mysql/logs
mkdir -p /kwang/soft/mysql/data
mkdir -p /kwang/soft/mysql/conf

```

```
# 安装MySQL
docker run --restart=always -p 3306:3306 --name mysql01 \
--net webnet --ip 172.18.0.2 \
-v /kwang/soft/mysql/logs:/var/log/mysql \
-v /kwang/soft/mysql/conf:/etc/mysql/conf.d \
-v /kwang/soft/mysql/data:/var/lib/mysql \
-e MYSQL_ROOT_PASSWORD=root \
-d mysql:8.0.29

# 连通网络
#docker network connect webnet mysql01
```

四、Elasticsearch 安装

```
# 1、下载镜像文件
docker pull elasticsearch:7.4.2
docker pull kibana:7.4.2            # 可视化界面

# 2、创建实例
mkdir -p /kwang/soft/elasticsearch/config
mkdir -p /kwang/soft/elasticsearch/data
mkdir -p /kwang/soft/elasticsearch/plugins
cd /kwang/soft/elasticsearch/config
echo "http.host: 0.0.0.0" >> /kwang/soft/elasticsearch/config/elasticsearch.yml

# 授权
chmod -R 777 /kwang/soft/elasticsearch

docker run --restart=always --name elasticsearch -p 9200:9200 -p 9300:9300 \
--net webnet --ip 172.18.0.6 \
-e "discovery.type=single-node" \
-e ES_JAVA_OPTS="-Xms64m -Xmx512m" \
-v /kwang/soft/elasticsearch/config/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml \
-v /kwang/soft/elasticsearch/data:/usr/share/elasticsearch/data \
-v /kwang/soft/elasticsearch/plugins:/usr/share/elasticsearch/plugins \
-d elasticsearch:7.4.2

# 验证
http:192.168.190.128:9200

# 3、安装Kibana
docker run --restart=always \
--name kibana \
--net webnet --ip 172.18.0.7 \
-e ELASTICSEARCH_URL=http://172.18.0.6:9200 \
-p 5601:5601 \
-d kibana:7.4.2

# 进入容器，修改kibana.yml中 elasticsearch.hosts 为容器ip
elasticsearch.hosts: [ "http://172.18.0.6:9200" ]

# 验证
http:192.168.190.128:5601

# 连通网络
#docker network connect webnet elasticsearch
```

五、Nginx 安装

```
docker run -p 80:80 --name nginx -d nginx
# 复制配置文件
mkdir /kwang/soft/nginx
cd /kwang/soft/nginx
docker container cp nginx:/etc/nginx .
mv nginx conf

# 停止并删除Nginx
docker stop nginx
docker rm nginx

docker run --restart=always -p 80:80 --name nginx \
--net webnet --ip 172.18.0.3 \
-v /kwang/soft/nginx/html:/usr/share/nginx/html \
-v /kwang/soft/nginx/logs:/var/log/nginx \
-v /kwang/soft/nginx/conf:/etc/nginx \
-d nginx

# 连通网络
#docker network connect webnet nginx
```

踩坑记录： 记得每个配置项最后加";"

六、Nacos安装

```
# 创建目录
mkdir -p /kwang/soft/nacos/logs
mkdir -p /kwang/soft/nacos/conf

# 初始化脚本
```

```
https://github.com/alibaba/nacos/blob/master/config/src/main/resources/META-INF/nacos-db.sql

# 安装nacos
docker  run \
--name nacos -d \
-p 8848:8848 \
-e MODE=standalone \
nacos/nacos-server

# 将配置文件拷入本地
docker cp nacos:/home/nacos/conf /kwang/soft/nacos

# 停止并删除nacos
docker stop nacos
docker rm nacos

# 安装 nacos
docker  run -d \
--restart=always \
--privileged=true \
--net webnet --ip 172.18.0.4 \
--name nacos \
-p 8848:8848 \
-p 9848:9848 \
-p 9849:9849 \
-e JVM_XMS=256m \
-e JVM_XMX=256m \
-e MODE=standalone \
-v /kwang/soft/nacos/logs:/home/nacos/logs \
-v /kwang/soft/nacos/data:/home/nacos/data \
-v /kwang/soft/nacos/conf:/home/nacos/conf \
nacos/nacos-server

# 连通网络
#docker network connect webnet nacos
```

踩坑记录

MySQL: 8.0.29

Nacos: 2.1.0

VM重启后频繁出现No DataSource set，查看日志后，在Nacos的application.properties的数据源链接中添加 `allowPublicKeyRetrieval=true` 后解决

七、Redis 安装

```
# 新建文件夹及文件
```

```
mkdir -p /kwang/soft/redis/conf
mkdir -p /kwang/soft/redis/data
touch /kwang/soft/redis/conf/redis.conf

# 修改配置文件
vi redis.conf
appendonly yes

# 运行redis
docker run -p 6379:6379 --name redis01 \
--restart=always \
--net webnet --ip 172.18.0.5 \
-v /kwang/soft/redis/data:/data \
-v /kwang/soft/redis/conf/redis.conf:/etc/redis/redis.conf \
-d redis redis-server /etc/redis/redis.conf
```

八、RabbitMQ 安装

```
# 新建文件夹
mkdir -p /kwang/soft/rabbitmq/data

# 运行
docker run -d --restart=always --name rabbitmq \
-p 5672:5672 -p 15672:15672 \
-v /kwang/soft/rabbitmq/data:/var/lib/rabbitmq \
--hostname myRabbit \
-e RABBITMQ_DEFAULT_VHOST=my_vhost \
-e RABBITMQ_DEFAULT_USER=admin \
-e RABBITMQ_DEFAULT_PASS=admin \
rabbitmq:3.10-management

#
http://106.52.72.75:15672/
```