6-9 | Docker-Compose 容器集群化管理的介绍

如果没有集群管理工具，我们的 Docker 集群要如何进行部署？

1. docker run -p 9091:9091 --name qiyu-live-user-provider-01 -d --add-host 'qiyu.nacos.com:127.0.0.1' -e 'jvm 参数内容' registry.baidubce.com/qiyu-live-test/qiyu-live-user-provider-docker:1.0.2

1. docker run -p 9092:9092 --name qiyu-live-user-provider-02 -d --add-host 'qiyu.nacos.com:127.0.0.1' -e 'jvm 参数内容' registry.baidubce.com/qiyu-live-test/qiyu-live-user-provider-docker:1.0.2

1. docker run -p 9093:9093 --name qiyu-live-user-provider-03 -d --add-host 'qiyu.nacos.com:127.0.0.1' -e 'jvm 参数内容' registry.baidubce.com/qiyu-live-test/qiyu-live-user-provider-docker:1.0.2

什么是 Docker-Compose

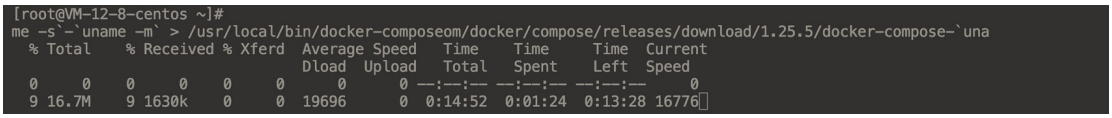
Compose 是用于定义和运行多容器 Docker 应用程序的工具。通过 Compose，您可以使用 YML 文件来配置应用程序需要的所有服务。然后，使用一个命令，就可以从 YML 文件配置中创建并启动所有服务。

关于 Docker-Compose 的安装

执行命令：

```
SQL
 curl -L
https://github.com/docker/compose/releases/download/1.25.5/docker-compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose
```

下载 github 上的 docker-compose 文件，执行命令后，如下图所示：



将可执行权限应用于二进制文件：

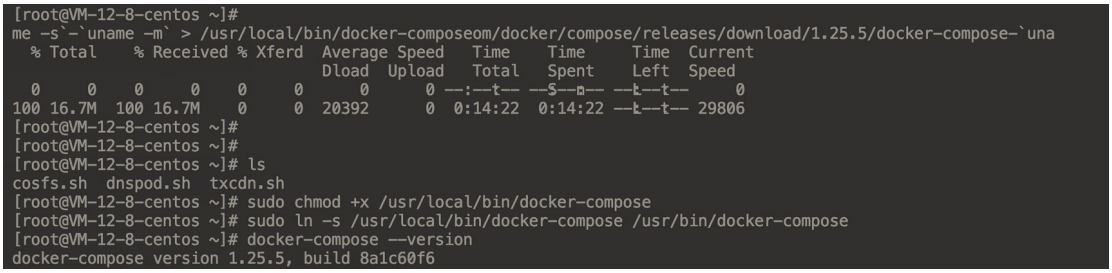
```
SQL
$ sudo chmod +x /usr/local/bin/docker-compose
```

创建软链：

```
SQL
$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

测试是否安装成功：

```
SQL
$ docker-compose --version
```



编写基于 user-provider 的 docker-compose 文件

编写 docker-compose.yml 文件，内容如下：

```
SQL
version: '3'
services:
  qiyu-live-user-provider-docker-1:
    container_name: qiyu-live-user-provider-docker-1
    image: 'registry.baidubce.com/qiyu-live-test/qiyu-live-user-
provider-docker:1.0.4'
    ports:
      - "9091:9091"
    #注意 xss 参数是设置线程的栈空间大小，这里至少要比 136k 大，否则会出
错
    environment:
      - DUBBO_IP_TO_REGISTRY=192.168.1.7
      - DUBBO_PORT_TO_REGISTRY=9091
```

```
      - DUBBO_PORT_TO_BIND=9091
      - JAVA_OPTS=-XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=128m
-Xms512m -Xmx512m -Xmn128m -Xss256k
    extra_hosts:
      - 'qiyu.nacos.com:175.178.130.183'
      - 'cloud.db:175.178.130.183'
      - 'qiyu.rmq.com:192.168.40.167'

  qiyu-live-user-provider-docker-2:
    container_name: qiyu-live-user-provider-docker-2
    image: 'registry.baidubce.com/qiyu-live-test/qiyu-live-user-
provider-docker:1.0.4'
    ports:
      - "9092:9092"
    #注意 xss 参数是设置线程的栈空间大小，这里至少要比 136k 大，否则会出
错
    environment:
      - DUBBO_IP_TO_REGISTRY=192.168.1.7
      - DUBBO_PORT_TO_REGISTRY=9092
      - DUBBO_PORT_TO_BIND=9092
      - JAVA_OPTS=-XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=128m
-Xms512m -Xmx512m -Xmn128m -Xss256k
    extra_hosts:
      - 'qiyu.nacos.com:175.178.130.183'
      - 'cloud.db:175.178.130.183'
      - 'qiyu.rmq.com:192.168.40.167'
```

在文件里面注入我们的 host 地址，和镜像文件名称，容器内部的端口号，JVM 启动参数，部署节点几台

由于我们是集群部署，所以启动的时候一定不能指定宿主机的端口号，否则会出错。

启动的命令如下：

```SQL
docker-compose -f ./docker/docker-compose.yml up
```

如果需要后台启动，则加入-d 命令即可：

```SQL
docker-compose -f ./docker/docker-compose.yml up  -d
```

编写基于 live-api 的 docker-compose 文件

```yaml
version: '3'
services:
  qiyu-live-api-docker-1:
    container_name: qiyu-live-api-docker-1
    image: 'registry.baidubce.com/qiyu-live-test/qiyu-live-api-
docker:1.0.4'
    ports:
      - "8081:8081"
    environment:
      - server.port=8081
      - JAVA_OPTS=-XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=128m
-Xms512m -Xmx512m -Xmn128m -Xss256k
    extra_hosts:
      - 'qiyu.nacos.com:175.178.130.183'
  qiyu-live-api-docker-2:
    container_name: qiyu-live-api-docker-2
    image: 'registry.baidubce.com/qiyu-live-test/qiyu-live-api-
docker:1.0.4'
    ports:
      - "8082:8082"
    environment:
      - server.port=8082
      - JAVA_OPTS=-XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=128m
-Xms512m -Xmx512m -Xmn128m -Xss256k
    extra_hosts:
      - 'qiyu.nacos.com:175.178.130.183'
```