

Guide for YARN-Docker Image

Shangyu XIE

January 3, 2017

1 Instruction

1.1 Background

This guide is to explore the Hadoop-docker image via SEQUENCEIQ, referring to the following websites.

- *Docker Container as YARN Container* via reusing the docker on host.
- Github code of Hadoop-docker 2.6.0, of which the tag is *dce*.
- Github code of Hadoop-docker 2.7.0, which is the pseudo-distributed mode of Hadoop.

1.2 Note

- a. Docker Container Executor(DCE) initially work for Hadoop-yarn to launch docker container via configuration for yarn-site.xml.
- b. The 2.7.0 version just simply present a docker image which has Hadoop 2.7.0, and there is nothing about docker inside. It can be used as the docker container image.

2 Analysis

2.1 Working Mechanisms

Comparing the dce and 2.7.0 version image, you can find the following commons and differences:

- a. The Hadoop configuration file, yarn-site.xml, is added the configuration properties of docker.
- b. Both initialize the Hadoop pseudo-distributed mode when the containers initializes.

And from the running command,

```
docker run -i -t -v /usr/local/bin/docker:/usr/local/bin/docker
-v /var/run/docker.sock:/var/run/docker.sock
-v /tmp/hadoop-root/nm-local-dir:/tmp/hadoop-root/nm-local-dir
-v /usr/local/hadoop/logs/userlogs:/usr/local/hadoop/logs/userlogs
-net=host sequenceiq/hadoop-docker:2.6.0-dce /etc/bootstrap.sh -bash
```

We can get two points,

- Mount the four files directory, and the first two are for docker, while the latter two are for Hadoop logs.
- *net=host* means that the container binds the ports on host, from which we can get that this so-called new plan for DCE is just using a docker container as agent for host system. Furthermore, it does not mean "container in container"; in other words, you can imagine that a "master" container initializes multiple "slave" or "child" containers work for Hadoop on host system, where the platform is still host's docker daemon.

2.2 Modification

You may notice that the docker path on guide, */usr/local/bin/docker* is not consistent with the default. You can use the command:

```
$ which docker
```

And the output is:

```
/usr/bin/docker
```

So just modify it, otherwise the container cannot work well.

What's more, you should run

```
$ docker rm -v CONTAINER_NAME
```

to delete the container. The addition of *-v* is for the remove of mount files directory. Because the data volume is independent of container itself, that is, you cannot delete it via simply deleting docker container. More details for here.

3 Test

3.1 Docker

There should be a error for docker if you use the default image file(In fact, you can use any hadoop-docker image for docker test as long as modifying the yarn-site.xml file).

So for testing docker, just run

```
$ docker run -it -v usr/bin/docker:/usr/bin/docker
-v /var/run/docker.sock:/var/run/docker.sock
-net=host sequenceiq/hadoop-docker:2.6.0-dce /etc/bootstrap.sh -bash
```

After running this, you can see

- `$ jps.` outputs well, the processes of hadoop all display.

- `$ docker ps`. it goes wrong.

The error is obvious to understand.

docker: error while loading shared libraries: libltdl.so.7: cannot open shared object file: No such file or directory

There is an open issue on Github. In short, it lack the dependency for docker in container itself. So you can solve this via add the docker binary code(install the docker, and just mount the docker.sock).

And just use the docker install guide on CentOS, you will find the version of docker image (version:6.8) is too old for docker(required: 7). However, it can install docker. It outputs as you wish

`$ docker ps`

It outputs all the running container on host.

`$ docker images`

It will list all the images downloaded on host.

3.2 Hadoop

Run Teragen example.

`hadoop jar $ HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar teragen`

`-Dmapreduce.map.env="yarn.nodemanager.docker-container-executor.image-name=sequenceiq/hadoop-docker:2.7.1"`

`-Dmapreduce.reduce.env="yarn.nodemanager.docker-container-executor.image-name=sequenceiq/hadoop-docker:2.7.1"`

`-Dyarn.app.mapreduce.am.env="yarn.nodemanager.docker-container-executor.image-name=sequenceiq/hadoop-docker:2.7.1" 1000 output`

There will be an error as below,

ExitCodeException exitCode=1: Error: No such image, container or task: containerXXXXXXXXXX

It is kind of weird because all the images are downloaded already, so there is just one possible reason, that docker in container cannot find the correct the images.

Another problem, is that hadoop just gets stuck here.