

Hippo单机版部署手册

Prerequisite

配置文件准备

容器网络部署

docker compose配置

磁盘路径配置

启停服务

主机网络模式

docker compose配置

IP配置

端口配置

master group配置

MISC

更改容器内日志目录

更改容器内配置文件目录

JVM堆参数配置

服务端JVM子进程配置(仅scope、argodb涉及)

本文档介绍如何基于docker-compose部署hippo单机版。

Prerequisite

- 安装docker-compose, <https://docs.docker.com/compose/reference/>
- 本地已下载tddms-standalone image
 - 本文档中, 将假设image路径为transwarp/tddms-standalone:develop

配置文件准备

从tddms-standalone image镜像中，将/usr/local/standalone-template目录拷贝到本地，本例假设本地目录为/root/standalone-template。

```
Shell | 复制代码  
1 docker cp $(docker create --name tc transwarp/tddms-standalone:develop):/usr/local/standalone-template . && docker rm tc
```

cd进standalone-template目录，将看到以下目录结构：

```
Shell | 复制代码  
1 [root@tw-node46 example]# ll  
2 total 8  
3 drwxr-xr-x 3 root root 4096 Jul 12 19:39 conf  
4 -rw-r--r-- 1 root root 591 Jul 12 19:39 docker-compose.yml  
5 drwxr-xr-x 2 root root 6 Jul 12 19:39 log
```

其中，conf目录下是hippo需要的配置文件；docker-compose.yml为为docker compose service配置文件；log目录是一个空目录，仅为挂载日志目录使用。

容器网络部署

容器网络部署是最简单、最常见的单机版部署模式，也是standalone-template里展示的模式。

docker compose配置

容器网络的docker-compose.yml内容如下：

```
1 version: "3"
2 services:
3   hippo:
4     ports:
5       - "7788:18902"
6       - "7789:18822"
7     volumes:
8       - /mnt/disk1/hippo-standalone-master/data:/vdir/mnt/disk1/shiva-master-hippo/data/
9       - /mnt/disk1/hippo-standalone-tserver/data:/vdir/mnt/disk1/shiva-tabletserver-hippo/data
10      - /mnt/disk2/hippo-standalone-tserver/data:/vdir/mnt/disk2/shiva-tabletserver-hippo/data
11      - /root/standalone-template/log:/var/log/hippo
12      - /root/standalone-template/conf:/etc/hippo/conf
13     environment:
14       CONF_DIR: "/etc/hippo/conf"
15       LOG_DIR: "/var/log/hippo"
16     image: "transwarp/tddms-standalone:develop"
```

该配置文件定了一个名为hippo的service，关键部分如下：

- ports
 - 将hippo http server与webserver的容器内端口映射到容器外。
 - 18902为http server端口，需要与conf/elasticsearch.yml里定义匹配。
 - 18822为webserver端口，需要与conf/shiva-restful.sh里定义匹配。
- volumes
 - 第一行，定义master数据目录挂载，将本地目录/mnt/disk1/hippo-standalone-tserver/data挂载到容器的/vdir/mnt/disk1/shiva-tabletserver-hippo/data。注意，在第一次部署时，需要手工创建本地目录。
 - 第二、三行，定义tserver数据目录挂载，本例中tserver使用disk1和disk2两块磁盘，分别挂载到容器的/vdir/mnt/disk1/shiva-tabletserver-hippo/data和/vdir/mnt/disk2/shiva-tabletserver-hippo/data目录。注意，在第一次部署时，需要手工创建本地目录。
 - 第四行挂载容器内日志目录，如果希望更换本地日志目录，可按需调整；默认保留在standalone-template的log目录下。
 - 第五行挂载容器内配置目录，standalone-template/conf目录下是hippo需要的配置文件。
 - 注意，master、tserver的数据目录，需要与conf下配置文件中内容匹配，详见下节。
- environment

- CONF_DIR和LOG_DIR环境变量，需要与volumes里配置、目录的容器内挂载点对应。
- image
 - 定义service使用的image，如果镜像名不是“transwarp/tddms-standalone:develop”，请按真实情况调整。

磁盘路径配置

在容器网络部署模式下，只需要为master、tserver配置磁盘路径。

- master.conf配置文件，“data_path=/vdir/mnt/disk1/shiva-master-hippo/data”，表示master数据将存储在容器内的/vdir/mnt/disk1/shiva-master-hippo/data路径。注意，此处必须与docker compose配置文件的挂载路径对应(volumes节)。master只使用一个磁盘路径。
- store.conf配置文件配置tserver使用的磁盘路径，默认模板里定义了两个store，store的data_dir配置表示容器内路径。注意，此处同样必须与docker compose配置文件的挂载路径对应(volumes节)。

启停服务

当完成关键配置后，在standalone-template目录下通过docker-compose up既可以启动服务，当输出“Hippo standalone start successfully”后，服务启动完成。

通过docker-compose up -d以Detached模式后台启动。

通过docker-compose ps查看服务状态。

通过docker-compose stop hippo停止服务。

主机网络模式

容器模式部署有一些限制：

- 后续无法扩容成集群模式
- 无法使用hippo跨集群相关功能，包括
 - 跨集群copy by query
 - 跨集群ccr

如果这些限制在您的应用场景中无法接受，可以使用主机网络模式进行部署。如果使用主机网络部署，除了配置磁盘路径外，您还需要配置hippo服务的ip、端口以及master group。

docker compose配置

其中docker-compose.yml为docker compose service配置文件，内容如下：

```
1  version: "3"
2  services:
3    hippo:
4      network_mode: "host"
5      volumes:
6        - /mnt/disk1/hippo-standalone-master/data:/vdir/mnt/disk1/shiva-master-hippo/data/
7        - /mnt/disk1/hippo-standalone-tserver/data:/vdir/mnt/disk1/shiva-tabletserver-hippo/data
8        - /mnt/disk2/hippo-standalone-tserver/data:/vdir/mnt/disk2/shiva-tabletserver-hippo/data
9        - /root/standalone-template/log:/var/log/hippo
10       - /root/standalone-template/conf:/etc/hippo/conf
11     environment:
12       CONF_DIR: "/etc/hippo/conf"
13       LOG_DIR: "/var/log/hippo"
14     image: "transwarp/tddms-standalone:develop"
```

与之前区别在于：

- network_mode, 设置为host, 使用主机模式

IP配置

需要在以下两个配置文件中，配置hippo使用的ip信息，本示例假设使用10.0.0.46。

- master.conf配置文件，配置public_host=10.0.0.46
- topology.conf配置文件，配置public_host=10.0.0.46

端口配置

需要为master、tserver、http server以及webserver配置端口。

- master端口，在master.conf配置文件中，配置"master_service_port=18650"，表示master将使用从18650开始的连续四个端口。
- tserver端口，在tabletserver.conf中，配置"manage_service_port=18702"，表示tserver将使用从18702开始的连续四个端口。

- http server端口，在elasticsearch.yml配置文件中，配置"http.port: 18902"，表示http server将监听在18902端口。
- webserver端口，在shiva-restful.sh配置文件中，"-Dhttp_port=18822"，表示webserver将监听在18822端口。

master group配置

在shiva-env.sh中，配置“export MASTER_GROUP=10.0.0.46:18650”，master group其实是master的服务地址，因为是standalone，其实该配置就是IP加master端口。

MISC

更改容器内日志目录

standalone模板中，容器内日志挂载目录为/var/log/hippo，如果希望修改这个配置(虽然我很好奇这样做的动机)，可以通过修改如下配置完成：

- 修改docker-compose.yml中，volumes与environment，日志目录对应项。
- 修改master.flags配置文件中，--log_dir配置项。
- 修改log4j2.xml.webserver配置文件中，log4j的fileName、filePattern配置。
- 修改search.yml配置文件中，path.logs配置项。
- 修改jvm.options配置文件中，HeapDumpPath与ErrorFile配置。
- 修改log4j2.xml.tserver配置文件中，log4j的fileName、filePattern配置。
- 修改log4j2.xml.httpsrver配置文件中，log4j的fileName、filePattern配置。

更改容器内配置文件目录

standalone模板中，容器内配置文件挂载目录为/etc/hippo/conf，如果希望修改这个配置(同样，我很好奇这样做的动机)，可以通过修改如下配置完成：

- 修改docker-compose.yml中，volumes与environment，配置目录对应项。
- 修改tabletserver.conf中
 - topology_conf
 - store_conf

- dynamic_config_path
- jvm_options_file
- log4j_file
- search_conf_file
- plugin_path

JVM堆参数配置

hippo standalone模式会启动一个http server、一个webserver进程，http server与webserver是java进程，在节点内存受限的情况下，可能需要调整jvm堆大小。

- http server的堆大小在shiva-http-server.sh文件配置(见JAVA_OPTS)，http server堆大小不宜过小，推荐最少分配8GB，默认31GB。
- webserver的堆大小在shiva-restful.sh文件配置(通用在JAVA_OPTS)。

服务端JVM子进程配置(仅scope、argodb涉及)

scope与argodb均会在tserver启动子进程，子进程中的jvm启动参数由jvm.options文件配置，其中堆大小默认31GB。