

数据备份和恢复

本篇是第五部分“存储篇”的第二篇，上一篇我为你介绍了 Docker volume 的生命周期管理相关的内容，本篇我将为你介绍基于 Docker volume 的数据备份和恢复。

我们在使用 Docker 时，难免会有数据备份和恢复相关的需求，尤其是处理与数据库等有状态的应用有关的问题时。

一般情况下，我们可能会选择将数据先备份至硬盘上，再拷贝至其他目标位置，最后进行数据恢复。

另外一种选择则是直接使用分布式文件存储，数据直接备份在分布式文件存储中，目标位置直接挂载即可进行数据恢复。

上篇我为你介绍了 Docker volume，对于我们使用而言，Docker volume 完全由 Docker 进行管理，只需要挂载即可使用。使用中可以减少很多类似数据拷贝之类的繁琐工作。

下面我们以 Redis 容器的数据备份恢复为例。

准备配置文件

如果不为 Redis 提供配置文件的话，则它会使用内置的默认配置。这里我们为它提供一个配置文件，指定数据存储位置。

这个配置文件，我们直接将其放入一个新创建的 volume 中。

- 创建 volume 存储配置文件

```
(MoeLove) → ~ docker volume create redis-conf  
redis-conf
```

[复制](#)

- 写配置文件

复制

```
(MoeLove) → ~ docker run --rm -it --mount src=redis-conf,dst=/redis-conf alpine
/ # cat <<'EOF' > /redis-conf/redis.conf
dir /data
save 900 1
save 300 10
save 60 10000
EOF
/ # cat /redis-conf/redis.conf
dir /data
```

- exit 退出容器

验证数据备份和恢复流程

- 创建 volume 用于存放数据

复制

```
(MoeLove) → ~ docker volume create redis-data
redis-data
```

- 启动 Redis 容器，挂载配置文件和数据存储的 volume

复制

```
(MoeLove) → ~ docker run --rm --mount src=redis-conf,dst=/redis-conf --mount src
1:C 22 Jan 2020 08:21:42.623 # o000o000o000o Redis is starting o000o000o000o
1:C 22 Jan 2020 08:21:42.623 # Redis version=5.0.7, bits=64, commit=00000000, modi
1:C 22 Jan 2020 08:21:42.623 # Configuration loaded
1:M 22 Jan 2020 08:21:42.625 * Running mode=standalone, port=6379.
1:M 22 Jan 2020 08:21:42.625 # Server initialized
1:M 22 Jan 2020 08:21:42.625 # WARNING overcommit_memory is set to 0! Background s
1:M 22 Jan 2020 08:21:42.625 # WARNING you have Transparent Huge Pages (THP) suppo
1:M 22 Jan 2020 08:21:42.625 * Ready to accept connections
```

- 连接 Redis server 并写入数据

复制

```
(MoeLove) → ~ docker exec -it $(docker ps -ql) redis-cli
127.0.0.1:6379> info keyspace
# Keyspace
127.0.0.1:6379> set name docker-volume
OK
```

- 主动触发数据持久化，进行备份操作

复制

```
(MoeLove) → ~ docker exec -it $(docker ps -ql) redis-cli
127.0.0.1:6379> save
OK
```

我们刚才的配置文件中是有配置 `save` 相关参数的，这里为了演示，主动执行 `save` 操作触发持久化。这样 Redis 就会将数据写入到刚才配置 `/data/dump.rdb` 文件中。

现在我们停止此容器：

复制

```
(MoeLove) → ~ docker exec -it $(docker ps -ql) redis-cli shutdown
```

- 验证数据恢复

重新启动一个容器，挂载刚才的 volume 验证下数据的恢复：

复制

```
(MoeLove) → ~ docker run -d --rm --mount src=redis-conf,dst=/redis-conf --mount
f2769d0925ab7c6a77d6fd257e780711787f75fef5c051ded02f88cd73fff82b
(MoeLove) → ~ docker exec -it $(docker ps -ql) redis-cli
127.0.0.1:6379> info keyspace
# Keyspace
db0:keys=1,expires=0,avg_ttl=0
127.0.0.1:6379> get name
"docker-volume"
```

可以看到，刚才的数据已经恢复到了这个新的 Redis 实例中了。

总结

本篇，我为你介绍了如何使用 Docker volume 进行数据备份和恢复，当然也可以有其他的一些操作方式，比如：

复制

```
# 查看 volume
(MoeLove) → ~ docker volume ls
DRIVER          VOLUME NAME
local           redis-conf
local           redis-data

# 查看 volume 的详细信息
(MoeLove) → ~ docker volume inspect redis-data
[
  {
    "CreatedAt": "2020-01-22T16:26:09+08:00",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/redis-data/_data",
    "Name": "redis-data",
    "Options": {},
    "Scope": "local"
  }
]
```

在 volume 的详细信息中，可以看到其在主机上的具体位置。

复制

```
(MoeLove) → ~ sudo ls /var/lib/docker/volumes/redis-data/_data
dump.rdb
```

其中包含了我们所需的具体数据文件，你可以看具体需求来使用这些数据文件。

下一篇，我将为你深入介绍 Docker 的存储驱动 Overlay2，带你理解 Docker 的存储原理。