

关于如何在红帽下配置devicemapper

http://blog.csdn.net/qq_26923057/article/details/52351731

```
1 <span data-wiz-span="data-wiz-span" style="font-size: 0.833rem;">systemctl stop docker
2
3 pvcreate /dev/vdb # 创建PV，一般一个盘相当于一个PV
4 vgcreate docker /dev/vdb #创建VG，一个VG相当于一个存储池，多个PV可以加入一个VG
5 vgdisplay docker
6 lvcreate --wipesignatures y -n thinpool docker -l 95%VG # 创建逻辑卷thinpool，占docker这个VG的95%
7 lvcreate --wipesignatures y -n thinpoolmeta docker -l 1%VG # 创建逻辑卷thinpoolmeta，占docker这个VG的1%
8 lvconvert -y --zero n -c 512K --thinpool docker/thinpool --poolmetadata docker/thinpoolmeta # 将thinpool转变成精简卷，poolme
9
10 # 配置自动扩容
11 cat <<EOF > /etc/lvm/profile/docker-thinpool.profile
12 activation {
13
14     thin_pool_autoextend_threshold=80
15     thin_pool_autoextend_percent=20
16
17 }
18 EOF
19 # 应用配置
20 lvchange --metadataprofile docker-thinpool docker/thinpool
21 lvs -o+seg_monitor
22
23 # 配置docker
24 vi /etc/sysconfig/docker-storage
25 DOCKER_STORAGE_OPTIONS= --storage-driver=devicemapper --storage-opt dm.thinpooldev=/dev/mapper/docker-thinpool --storage-op
26
27 # 删除原来的数据???
28 rm -rf /var/lib/docker/[graphdriver]
29 systemctl daemon-reload
30 systemctl start docker</span>
```

官方关于devicemapper的介绍

devicemapper不适合密集IO的场景，并且loop-lvm模式不适合生产，只能用来测试

<https://docs.docker.com/engine/userguide/storagedriver/device-mapper-driver/>

关于loop设备：

<http://blog.csdn.net/11f021421/article/details/8871865>

关于安装docker

实际上docker自1.10版本起已经分为了社区版和企业版，因此安装新版的docker实际上是docker-ce。安装方式可以参看：

<https://docs.docker.com/engine/installation/linux/docker-ce/centos/#install-docker-ce-1>

关于docker-ce的配置文件：

<https://docs.docker.com/engine/reference/commandline/dockerd/#docker-runtime-execution-options>

关于docker API：

docker 源码中API使用swagger框架来管理，通过api/swagger.yaml，可以生成API Doc。

可以在swaggerhub或者官网上查看。

<https://app.swaggerhub.com/>

<https://docs.docker.com/engine/api/v1.32/>

关于代码结构:

api/swagger.yaml: Restful API的定义文件。

api/types/: 定义了Client和Server关于RestFul API的一些结构体定义

api/server: API server的实现

cli/: docker client的实现, 在cli/command目录下根据资源类型实现了Client的每一个命令, 比如: “cli/command/container/run.go”。

client/: 这个Client下的go文件应当负责和daemon进程交互。

daemon/: daemon进程的相关实现。

container: 容器对象的定义。

关于runC:

libcontainer的简易命令行接口, 相当virsh之于KVM。

<http://dockone.io/article/680>

<https://github.com/opencontainers/runc>

关于监控工具:

cadvisor、datadog、prometheus

```
1 docker run --volume=/:/rootfs:ro --volume=/var/run:/var/run:rw --volume=/sys:/sys:ro --volume=/var/lib/docker/:/var/lib/doc
2
```