



链滴

docker 系列 3--dockerd 配置文件及 dockerd 启动相关问题

作者: [james](#)

原文链接: <https://ld246.com/article/1520594009634>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

dockerd启动配置

docker通信方式选择

- docker默认以sock文件方式提供接口，要开放tcp接口远程调用，需要修改配置文件：
- The Docker daemon can listen for Docker Engine API requests via three different types of Socket: unix, tcp, and fd.参考：<https://docs.docker.com/engine/reference/commandline/docker/>
- You can configure the Docker daemon to listen to multiple sockets at the same time using multiple -H options:

```
# listen using the default unix socket, and on 2 specific IP addresses on this host.
```

```
$ sudo dockerd -H unix:///var/run/docker.sock -H tcp://192.168.59.106 -H tcp://10.10.10.2
```

配置文件daemon.json

- 使用配置文件：The daemon includes many configuration options, which you can pass as flags when starting Docker manually, or set in the **daemon.json** configuration file. The second method is recommended because those configuration changes persist when you restart Docker.
- 参考：[Configure and troubleshoot the Docker daemon](#)
- 参考文章：[docker深入2-linux下的配置文件daemon.json使用示例](#) 讲配置文件daemon.json的完整示例
- 如下运行的docker命令，可以用配置文件来配置：

```
$ dockerd -D --tls=true --tlscert=/var/docker/server.pem --tlskey=/var/docker/serverkey.pem  
-H tcp://192.168.59.3:2376  
$ sudo nohup docker -d --log-level warn -H tcp://0.0.0.0:2376 -H unix:///var/run/docker.sock  
> docker.log &  
{  
  "debug": true,  
  "tls": true,  
  "tlscert": "/var/docker/server.pem",  
  "tlskey": "/var/docker/serverkey.pem",  
  "hosts": ["unix:///var/run/docker.sock", "tcp://0.0.0.0:2375"]  
}
```

- 选择配置文件启动

```
--config string Location of client config files (default "/home/james/.docker")  
docker --config
```

docker启动方式

- 通过命令行启动docker，可能需要sudo：you can start Docker manually using the dockerd command. You may need to use sudo, depending on your operating system configuration.
- 可能会有权限不足的问题，如下

```
$ docker ps
```

```
Cannot connect to the Docker daemon. Is the docker daemon running on this host?  
E0307 06:52:52.025652 53165 container_manager.go:492] list containers[[running]] error: Get  
http://unix.sock/containers/json?filters=%7B%22status%22%3A%5B%22running%22%5D%7D:  
dial unix /var/run/docker.sock: connect: permission denied.
```

- 查看发现/var/run/docker.sock应该是属于docker用户组的，如果是root：root就会有错：

```
$ ll /var/run/docker.sock  
srw-rw----. 1 root docker 0 Mar  6 17:22 /var/run/docker.sock
```

- 参考：[docker技术注意点](#)，添加用户组
- 可以在docker启动时指定用户组：

```
sudo gpasswd -a ${USER} docker
```

```
$ dockerd -h | grep sock  
--containerd          Path to containerd socket  
-G, --group=docker    Group for the unix socket  
-H, --host=[]         Daemon socket(s) to connect to  
$ dockerd -G dockerroot --raw-logs
```

docker启动问题

- 在系统上yum安装了docker，手动启动了dockerd守护进程，但是发现运行时刻不能启动容器。有下错误：msg="Create container failed with error: shim error: docker-runc not installed on system"
m"
- 尝试链接docker-runc-current文件，但是没有解决问题。该方法参考：[重装docker后本身的容器启动失败](#)
- 在stackoverflow上看的如下问题：[docker-runc not installed on system](#)
，其中提到dockerd的systemd启动文件The service file located at /usr/lib/systemd/system/docker.service，这应该是yum安装的时候自动安装的启动文件。

```
--add-runtime docker-runc=/usr/libexec/docker/docker-runc-current \  
--default-runtime=docker-runc \  
--exec-opt native.cgroupdriver=systemd \  
--userland-proxy-path=/usr/libexec/docker/docker-proxy-current \  

```

- 使用systemctl来启动docker，这应该是推荐的启动方式：\$ sudo systemctl start docker.service
- 启动后，可以启动容器。但是自定义部分没有使用daemon.json的配置。
- 故参考《docker日志设置》里一篇对daemon.json的解析，在daemon.json里添加如下配置，再手启动，即可解决问题。

```
$ cat /etc/docker/daemon.json  
{  
  "log-level":"warn",  
  "hosts": ["unix:///var/run/docker.sock", "tcp://0.0.0.0:2375"],  
  "runtimes": {  
    "docker-runc": {  
      "path": "/usr/libexec/docker/docker-runc-current"  
    }  
  }  
}
```

```
},
"add-runtime": "docker-runc=/usr/libexec/docker/docker-runc-current",
"default-runtime": "docker-runc"
}
```

```
docker -d --log-level warn -H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock
```

docker启动选项

- 选择启动使用的根目录，默认是/var/lib/docker:

--data-root string	Root directory of persistent Docker state (default "/var/lib/doc er")
--pidfile string	Path to use for daemon PID file (default "/var/run/docker.pid")
-H, --host list	Daemon socket(s) to connect to (default [])

```
./dockerd --config-file daemon.json --raw-logs --log-level warn --data-root ./data/ --pidfile ./  
ocker.pid -H tcp://0.0.0.0:2376 -H unix:///home/bin/docker/docker-17.12.1/docker/docker.so  
k
```

docker升级

- docker升级操作，先安装新版docker
- 迁移镜像
- 停止老版docker，删除数据，容器和镜像
- 启动新版docker
- 容器原则上不存数据，故容器不需要迁移。仅需要迁移镜像。这里我们的镜像不多，一个个倒出倒把。
- 运行发现命令行可以运行容器，但是代码报错：

```
msg="Handler for POST /containers/create returned error: mkdir /mnt/resource/data/docker  
overlay2/4e32760280d0f8a6beefb2823a5c0534a11234e80165feae6bd4e5e7c0076d4c-init/me  
ged/dev/shm: invalid argument"
```

- 参考问题：[Cannt run or build Docker images on CentOS 7](#)

发现应该是storage-driver的问题。本地使用的存储引擎通过docker info查看：

```
$ docker info
Containers: 1
Running: 1
Paused: 0
Stopped: 0
Images: 39
Server Version: 17.12.1-ce
Storage Driver: overlay2
Backing Filesystem: extfs
Supports d_type: true
Native Overlay Diff: false
Logging Driver: json-file
```

Cgroup Driver: cgroupfs
Plugins:

```
$ uname -r  
3.10.0-327.13.1.el7.x86_64
```

- 同时参考: [Docker之几种storage-driver比较](#), 介绍OverlayFS在Linux3.18版本中被并入内核本机是3.10的可能不符合。
- 修改配置文件使用:

```
{  
  "log-level": "warn",  
  "hosts": ["unix:///var/run/docker.sock", "tcp://0.0.0.0:2375"],  
  "storage-driver": "devicemapper"  
}
```

安装docker

- 安装docker可以采用yum安装, 软件包安装, 或者直接使用二进制可执行文件。但是直接使用二进制可执行文件可能会有各种依赖问题, yum安装是最方便的方式。

yum安装docker遇到的问题

- 本来以为用yum安装是万无一失的, 但是还是出现了以下错误: 容器可以创建但是无法运行, 不映射端口可以运行, 映射端口就报错。说找不到docker-proxy。yum安装的也确实没有docker-proxy

```
Failed in starting container: API error (404): driver failed programming external connectivity on endpoint 1.ac_18_0_yztest-c_0-0-2_13_0_151a31de3de2cead (ea11474a0a341c7500d931e727d5f207c1be2d1d51158444aa101122dfad2b7): exec: "docker-proxy": executable file not found in $PATH.
```

```
$ docker run -d -p 9000:9000 --restart always -v /var/run/docker.sock:/var/run/docker.sock -v /opt/portainer:/data portainer/portainer  
4dc93965c584be10704ee5b0d0a1c14a1eabb1e2082a98095992065bd65dbf7b  
/usr/bin/docker-current: Error response from daemon: driver failed programming external connectivity on endpoint sleepy_boyd (99fda3a8fba386e88ef362b8f8059ea3d55fa734b353d6e6cba6c345e5b9a34): Bind for 0.0.0.0:9000 failed: port is already allocated.
```

- 通过将新版本的docker-proxy二进制文件放到/usr/bin目录下, 可以解决网络问题。