

Docker文档

一、Docker介绍

1. 官网: <https://www.docker.com/>
2. Doc: <https://docs.docker.com/>

二、Docker安装

1. 内核检查, Linux内核至少3.8以上。

```
1 sudo uname -a
```

2. 设置Docker存储库。

```
1 sudo yum install -y yum-utils
2 sudo yum-config-manager \
3   --add-repo \
4   https://download.docker.com/linux/centos/docker-ce.repo
```

3. 安装Docker引擎, 默认安装最新的。

```
1 sudo yum install docker-ce docker-ce-cli containerd.io
2 将用户加入docker组
3 sudo usermod -aG docker [用户名]
```

4. 设置Docker开机启动, 并启动Docker。

```
1 sudo systemctl enable docker && sudo systemctl start docker
```

5. 配置Docker国内镜像

```
1 sudo vi /etc/docker/daemon.json
2 {
3   "registry-mirrors": [
4     "https://v0i08fi4.mirror.aliyuncs.com",
5     "https://docker.mirrors.ustc.edu.cn",
6     "https://registry.docker-cn.com",
7     "https://hub-mirror.c.163.com"
8   ]
9 }
10 sudo systemctl daemon-reload && sudo systemctl restart docker
```

参考: <https://www.cnblogs.com/reasonzzy/p/11127359.html>

6. 其它命令

```
1 // Docker配置文件重启
2 sudo systemctl daemon-reload
3 // Docker重启
4 sudo systemctl restart docker
```

```
5 // Docker停止
6 sudo systemctl stop docker
7 // 查看Docker运行状态
8 sudo systemctl status docker
```

三、Docker镜像

1. 搜索镜像

```
1 docker search redis
```

2. 下载镜像

```
1 docker pull redis
```

3. 查看镜像列表

```
1 docker images
```

4. 删除镜像

```
1 docker rmi [镜像Id or 镜像名称]
```

5. 查看镜像信息

```
1 docker inspect redis
```

6. 查看镜像历史信息

```
1 docker history redis
```

7. 清理<none>镜像

```
1 docker rmi $(docker images -f "dangling=true" -q)
```

四、Docker容器

1. 运行容器

```
1 docker run --name my-redis -d -p 6379:6379 redis
```

2. 启动、停止、重启容器

```
1 docker start my-redis
2 docker stop my-redis
3 docker restart my-redis
```

3. 查看所有运行的容器

```
1 docker ps
```

4. 查看所有容器

```
1 docker ps -a
```

5. 查看容器信息

```
1 docker inspect my-redis
```

6. 进入容器内部

```
1 docker exec -it my-redis bash
```

7. 删除容器

```
1 docker rm -f my-redis
```

8. 更新容器配置

```
1 docker update --cpu-quota 1000000 my-redis
```

9. 查看容器日志

```
1 docker logs my-redis
```

五、Docker-Hub

网站地址：<https://www.docker.com/products/docker-hub>

1. 登录、登出

```
1 // 登录
2 docker login
3 // 登出
4 docker logout
```

六、Dockerfile

1. Docker指令

配置指令：

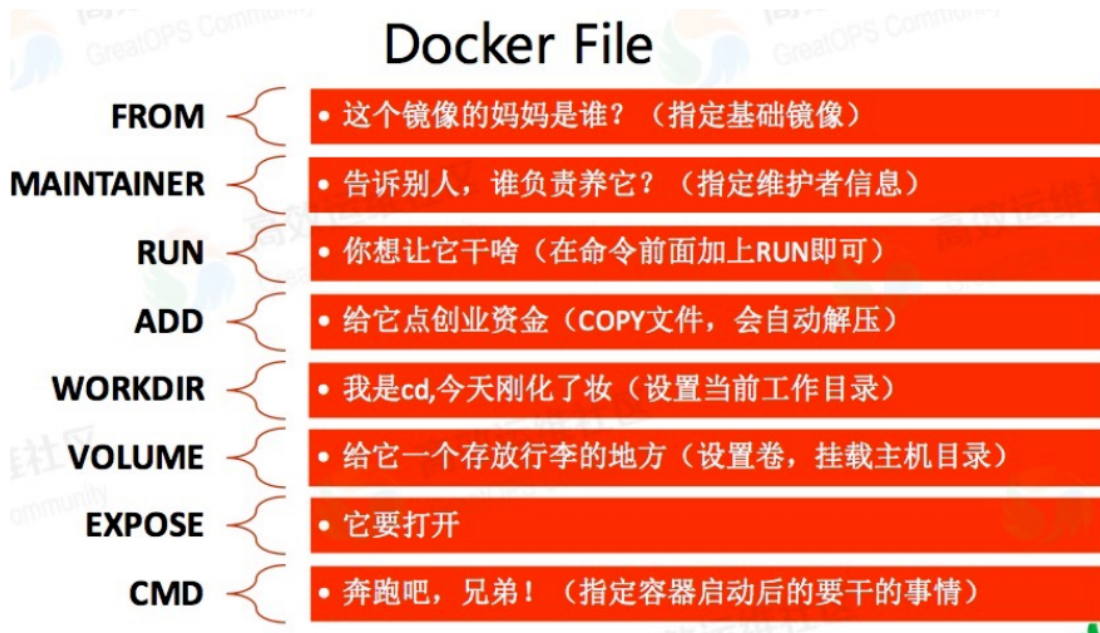
ARG	定义创建镜像过程中使用的变量
FROM	指定所创建镜像的基础镜像
LABEL	为生成的镜像添加元数据标签信息
EXPOSE	声明镜像内服务监听的端口
ENV	指定环境变量
ENTRYPOINT	指定镜像的默认入口命令
VOLUME	创建一个数据卷挂载点
USER	指定运行容器时的用户名或UID
WORKDIR	配置工作目录
ONBUILD	创建子镜像时指定自动执行的操作指令
STOPSIGNAL	指定退出的信号值
HEALTHCHECK	配置所启动容器如何进行健康检查
SHELL	指定默认shell类型

操作指令：

RUN 运行指定命令

CMD 启动容器时指定默认执行的命令
ADD 添加内容到镜像
COPY 复制内容到镜像

2. 全景图



3. 构建镜像

```
1 docker build -t easywxh/webapp:1.0 .
```

4. 上传镜像到Docker Hub

```
1 docker push easywxh/webapp:1.0
```

七、Docker volume

1. 创建数据卷

```
1 docker volume create my_vol
```

2. 查看数据卷列表

```
1 docker volume ls
```

3. 删除数据卷

```
1 docker volume rm my-vol
```

4. 查看数据卷信息

```
1 docker volume inspect my-vol
```

5. 清除无主数据卷

```
1 docker volume prune
```

6. 示例1: mysql挂载数据卷

```
1 docker run -d -p 3306:3306 --name dev-mysql -v /usr/local/dev/datadir:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=123123 mysql:5.7
```

7. 示例2: .net core挂载数据卷

```
1 docker build -t easywxh/volapp:1.0 .
2 docker run -it -p 7000:5000 --name myvolapp -v /usr/local/dev/voldata:/var/www easywxh/volapp:1.0 bash
```

八、Docker Compose

1. 安装Docker Compose

```
1 sudo curl -L "https://github.com/docker/compose/releases/download/1.26.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
2 sudo chmod +x /usr/local/bin/docker-compose
3 docker-compose --version
```

2. 构建

```
1 docker-compose build
2 #不带缓存的构建
3 docker-compose build --no-cache nginx
```

3. 部署Compose应用

```
1 docker-compose up -d
```

4. 停止容器并删除由创建的容器, 网络, 卷和镜像等

```
1 docker-compose down
```

5. 查看列表

```
1 docker-compose ps
```

6. 停止

```
1 docker-compose stop
```

7. 删除

```
1 docker-compose rm -f -v
```

8. 查看日志

```
1 docker-compose logs
2 docker-compose logs -f nginx
```

9. 重启已经停止的Compose应用

```
1 docker-compose restart
```

10. 登录到Compose应用容器中

```
1 docker-compose exec nginx bash
```

11. 获取一个命令的帮助

```
1 docker-compose 命令 --help
```

12. 查看各个服务容器内运行的进程

```
1 docker-compose top
```

13. Docker-Compose参考及准则

<https://docs.docker.com/compose/compose-file/>

九、Docker Swarm

1. 初始化

```
1 docker swarm init --advertise-addr 192.168.78.138:2377 --listen-addr 192.168.78.138:2377
```

2. 添加manager、worker节点

```
1 docker swarm join-token manager
2 docker swarm join-token worker
3 docker swarm join --token SWMTKN-1-69cn8las8j2sfu1rcwvdbv232qdawirljrtv1tq1et3us5h5em-7r37468i8262kc22n9rxe5ngv 192.168.78.138:2377 --advertise-addr 192.168.78.140:2377 --listen-addr 192.168.78.140:2377
4 docker swarm join --token SWMTKN-1-69cn8las8j2sfu1rcwvdbv232qdawirljrtv1tq1et3us5h5em-dgfa5fx9woxgl3n7tpdj9ux0m 192.168.78.138:2377 --advertise-addr 192.168.78.141:2377 --listen-addr 192.168.78.141:2377
```

3. 创建Service

```
1 docker service create --replicas 3 --name mywebapp -p 8080:5000 easywxh/webapp:1.0
```

4. 扩展服务

```
1 docker service update --replicas 5 mywebapp
```

5. 更新服务

```
1 docker service update --image easywxh/webapp:2.0 --update-parallelism 2 --update-delay 10s mywebapp
```

6. 查看服务列表

```
1 docker service ls <node>
```

7. 删除服务

```
1 docker service rm <node>
```

8. 节点退出集群

```
1 docker swarm leave -f #node
2 docker node rm -f <node> #manager
```

十、其它

1. docker node命令

```
1 docker node ls #查看所有集群节点
2 docker node rm #删除某个节点（-f强制删除）
3 docker node inspect ##查看节点详情
4 docker node demote #节点降级，由管理节点降级为工作节点
5 docker node promote #节点升级，由工作节点升级为管理节点
6 docker node update #更新节点
```

7 `docker node ps` #查看节点中的 Task 任务

2. docker service命令

```
1 docker service create #部署服务
2 docker service inspect #查看服务详情
3 docker service logs #查看某个服务日志
4 docker service ls #查看所有服务详情
5 docker service rm #删除某个服务 (-f强制删除)
6 docker service scale #设置某个服务个数
7 docker service update #更新某个服务
```

3. docker swarm命令

```
1 docker swarm init #初始化集群
2 docker swarm join-token worker #查看工作节点的 token
3 docker swarm join-token manager #查看管理节点的 token
4 docker swarm join #加入集群中
```

4. docker network命令

```
1 docker network connect #将容器连接到网络
2 docker network create #创建网络
3 docker network disconnect #断开容器与网络的连接
4 docker network inspect #查看网络信息
5 docker network ls #查看列表
6 docker network prune #删除所有未使用的网络
7 docker network rm #删除网络
```

5.