

## Docker镜像和容器的导入与导出

### docker镜像的导出和导入

显示当前docker中的镜像：

```
docker images
```

镜像列表如下：

| REPOSITORY | TAG    | IMAGE ID     | CREATED     | SIZE   |
|------------|--------|--------------|-------------|--------|
| pointsift  | latest | 90b2ef439b40 | 2 weeks ago | 12.6GB |
| ubuntu     | 18.04  | 735f80812f90 | 4 weeks ago | 83.5MB |

导出镜像：

```
docker save -o <保存路径> <镜像名称:标签>
```

如把A机 **ubuntu:18.04** 导出到当前文件夹，则在A机上运行：

```
docker save -o ./ubuntu18.tar ubuntu:18.04
```

此时，把A机当前文件夹下的**ubuntu18.tar**拷贝到另一台安装过**docker**的B机上，在B机上导入镜像：

```
docker load --input ./ubuntu18.tar
```

这样镜像**ubuntu:18.04**就成功的从A机复制到B机上了

### docker容器的导出与导入

显示当前docker中运行的容器：

```
docker ps
```

运行的容器列表如下：

| CONTAINER ID | IMAGE        | COMMAND     | CREATED       | STATUS |
|--------------|--------------|-------------|---------------|--------|
| PORTS        | NAMES        |             |               |        |
| 4a02996e83b1 | ubuntu:18.04 | "/bin/bash" | 44 secondsago | Up 42  |

seconds

ubuntu18

### 停止容器：

```
docker stop <容器名>
```

如要想要导出**ubuntu18**,必须先停止（如果**ubuntu18**没有运行，则不需要执行此步骤）：

```
docker stop ubuntu18
```

如果容器已经停止了，想要查看该容器，可以运行（该命令会显示所有的容器，包括运行的和非运行的）：

```
docker ps -a
```

### 导出容器：

```
docker export <容器名> > <保存路径>
```

### 如导在A机中导出容器**ubuntu18**:

```
docker export ubuntu18 > ./ubuntu18.tar
```

### 导入容器的命令：

```
docker import <文件路径> <容器名>
```

把A机当前文件夹下的**ubuntu18.tar**文件拷贝到B机上，在B机上运行：

```
docker import ./ubuntu18.tar ubuntu18
```

### 启动容器：

```
docker start <容器名>
```

B机上就多了一个名为**ubuntu18**的容器，但导入的容器还处在未运行的状态，需要启动容器：

```
docker start ubuntu18
```

## 进入容器：

#同一个容器会有相同的界面，如果该容器之前已经运行过，则会直接进入运行容器的界面

#该命令通常用于进入后台已经在运行的容器

`docker attach <容器名>`

或者

#同一个容器有不同的界面，进入容器后的界面不会受该容器的其他运行者的影响

#一般情况下建议使用该命令

`docker exec -it <容器名> <执行指令>`

## 启动容器之后，需要进入容器，可以运行：

`docker exec -it ubuntu18 /bin/bash`

容器导出后，使用Import导入后，应该是镜像吧，不是容器吧，`docker import ./ubuntu18.tar ubuntu18` 这里的ubuntu18 应该是导入后的镜像名吧，B机器里没有容器名为ubuntu18 的吧，如果想要启动容器，还需要通过镜像去运行吧