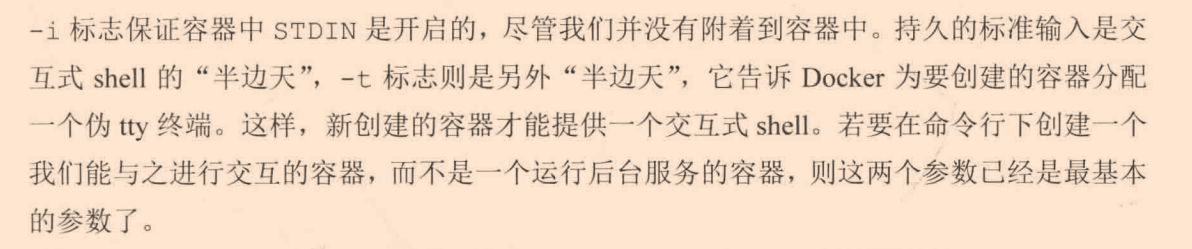
**运行容器： sodo docker run -i -t ubuntu /bin/bash**



**-d： 容器后台运行**

ubunte告诉docker基于ubuntu镜像来创建容器，然后在容器中运行/bin/bash启动了一个Bash shell

**exit推出shell界面，但是会导致容器停止运行**

**列出容器列表: docker ps （正在运行的）、 docker ps -a 所有容器列表**

**给容器命名： sodu docker run --name XXX -I -t ubuntu /bin/bash**

**创建一个容器并命名为XXX**

**重新启动已经停止的容器：sudo docker start (uuid、id、name都可)**

**启动容器后,只会在后台运行，不会跳出来命令行窗口需要我们用： docker attach**

**命令，重新附着到该容器的会话上。可以通过id等附着到指定容器 docker attach name**

**但是如果退出容器的shell，容器再次回停止运行**

**创建守护线程，命令与创建容器差不多，但-i -t 改为了-d 表示为后台运行。**

**sudo docker run --name XX -d ubuntu /bin/sh -c “while true; do echo hello world ; sleep 1; done”**

**docker logs dockername查看docker容器日志**

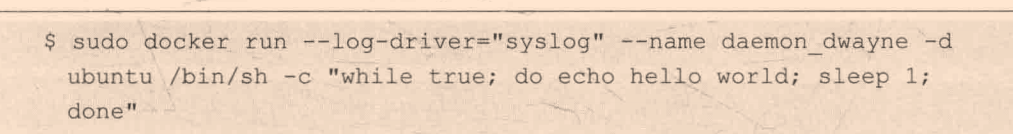
**docker logs -f dockername 与tail -f一致**

**docker logs --tail 10 dockername 查看最后10行**

**docker logs --tail 0 -f dockername 跟踪最新日志**

**也可以加-t 吧每个日志加上时间戳。docker logs -ft dockername**

**修改docker日志驱动，参数：--log-driver**



**查看容器内的进程：docker top dockername**

**docker 统计信息：docker stats dockername**

**docker中运行进程（后台任务）：docker exec -d dockername touch /erc/new\_config\_file**

**docker中运行进程（交互式任务）：docker exec -I -t dockername /bin/bash**

**停止进程：docker stop dockername**

**删除容器： docker rm dockername**

**查看镜像：docker images**

**下载镜像： docker pull imagesname**

**使用镜像名称+冒号+版本号启动不同版本镜像：docker run -i -t --name ubuntu\_12 ubuntu:12.04 /bin/bash**

**搜索镜像：docker search XX**