# Windows10安装Docker

## 先升级到win10专业版

## 任务管理器查看虚拟化是否开启

## 开启hyper

右键win--应用和功能—程序和功能—启动或关闭windows功能，选中Hyper-V

## 下载docker

<https://download.docker.com/win/stable/Docker%20for%20Windows%20Installer.exe>

## 安装完运行docker，看是否正常

可输入：

docker --version

docker-compose --version

docker-machine --version

docker ps

docker version

docker info

docker run hello-world

若出现问题：

could not read CA certificate的解决方法，说明之前安装过低版本的docker，环境变量中导入了docker virtualBox的变量，而该版本不需要导入这些变量，在环境变量中找到docker开头的删掉即可

# Windows开启Ubuntu子系统

背景：一般情况下，都是在linux上编译程序，所以在linux上跑一套代码是必要的，这时候linux也需要安装docker了，也会方便docker打包

参考连接：<https://jingyan.baidu.com/article/49711c61a1a025fa441b7cf2.html>

## 安装ubuntu系统

1. 应用和功能—程序和功能—启动或关闭windows功能，选中适用于linux的windows子系统
2. 在Microsoft store中搜索ubuntu，下载好后进入自己用户目录，切到root目录，初次进入root，需要sudo passwd设置密码

## 安装docker

1. 安装基本软件

$ sudo apt-get update  
$ sudo apt-get install apt-transport-https ca-certificates curl software-properties-common lrzsz -y

1. apt仓库添加并阿里云的docker软件包  
   # 把下载的key添加到本地trusted数据库中。  
   sudo curl -fsSL <https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg> | sudo apt-key add -  
   $ sudo add-apt-repository "deb [arch=amd64] <https://mirrors.aliyun.com/docker-ce/linux/ubuntu> $(lsb\_release -cs) stable"
2. 升级软件源  
   sudo apt-get update
3. 安装docker  
   sudo apt-get install docker-ce -y
4. 测试docker：docker version

安装依赖工具：sudo apt-get install python-pip -y  
安装编排工具：sudo pip install docker-compose  
查看版本：sudo docker-compose version

## 关联本地docker

1. 本地docker设置，勾选上本地
2. Ubuntu上輸入export DOCKER\_HOST=tcp://127.0.0.1:2375导入环境变量

# Docker基本命令

## 查看docker信息

docker version

docker info

## image操作

docker pull image\_name // 下载镜像

docker images // 镜像列表

docker rmi image\_name // 删除镜像

docker history image\_name //显示一个镜像的历史

## container操作

### 查看容器

docker ps

docker ps -a

docker ps -l

### 删除容器

docker rm `docker ps -a -q` // 删除所有终止的容器

docker rm $(docker ps -a -q) // 删除所有终止的容器

docker rm Name/ID // 删除单个容器

### 停止、启动、杀死一个容器

docker stop Name/ID

docker start Name/ID

docker kill Name/ID

docker restart Name/ID

# 微服务镜像打包部署

## 本地编写Dockerfile文件

# alpine打包  
FROM golang:alpine AS *builder*# 配置alpine使用清华源  
RUN echo -e "https://mirror.tuna.tsinghua.edu.cn/alpine/latest-stable/main\nhttps://mirror.tuna.tsinghua.edu.cn/alpine/latest-stable/community" > /etc/apk/repositories  
# 安装git  
RUN apk update && apk add --no-cache git gcc musl-dev  
# 设置iqiyi为工作目录（RUN执行的命令会在改目录下操作）  
WORKDIR $*GOPATH*/src/gitlab.com/cibn-kernel/mcp-micro/sync/iqiyi  
# 将当前目录（也就是执行docker build xxx 的目录）下的文件拷贝到项目根目录（对应的就是mcp-micro文件夹）下  
COPY . ../..  
# 配置docker镜像的系统环境：启用go modules  
ENV *GO111MODULE*=on  
# 配置go modules代理，package将从该开源代理拉取，避免GFW的问题  
ENV *GOPROXY*=https://goproxy.io  
# 获取所有package  
RUN go get -v  
# 构建项目，输出到镜像的指定目录下  
RUN GOOS=linux GOARCH=amd64 go build -ldflags="-w -s" -o /go/bin/iqiyi  
# 使用一个空的镜像打包发布后的go项目，以达到镜像体积的最小化  
FROM alpine  
# 从上面的镜像中拷贝编译后的程序到当前镜像的指定位置  
COPY --from=*builder* /go/bin/iqiyi /go/bin/iqiyi  
# 将config文件拷贝到编译文件的同目录下  
COPY --from=*builder* /go/src/gitlab.com/cibn-kernel/mcp-micro/sync/iqiyi/conf /go/bin/conf  
# 暴露镜像的8881端口  
EXPOSE 10008  
# 运行  
ENTRYPOINT ["./go/bin/iqiyi"]  
  
# -f 参数用来指定Dockerfile文件的位置  
# docker build -t iqiyi -f ./sync/iqiyi/Dockerfile .

## 执行dockerfile文件一键打包镜像

docker build -t iqiyi -f ./sync/iqiyi/Dockerfile .

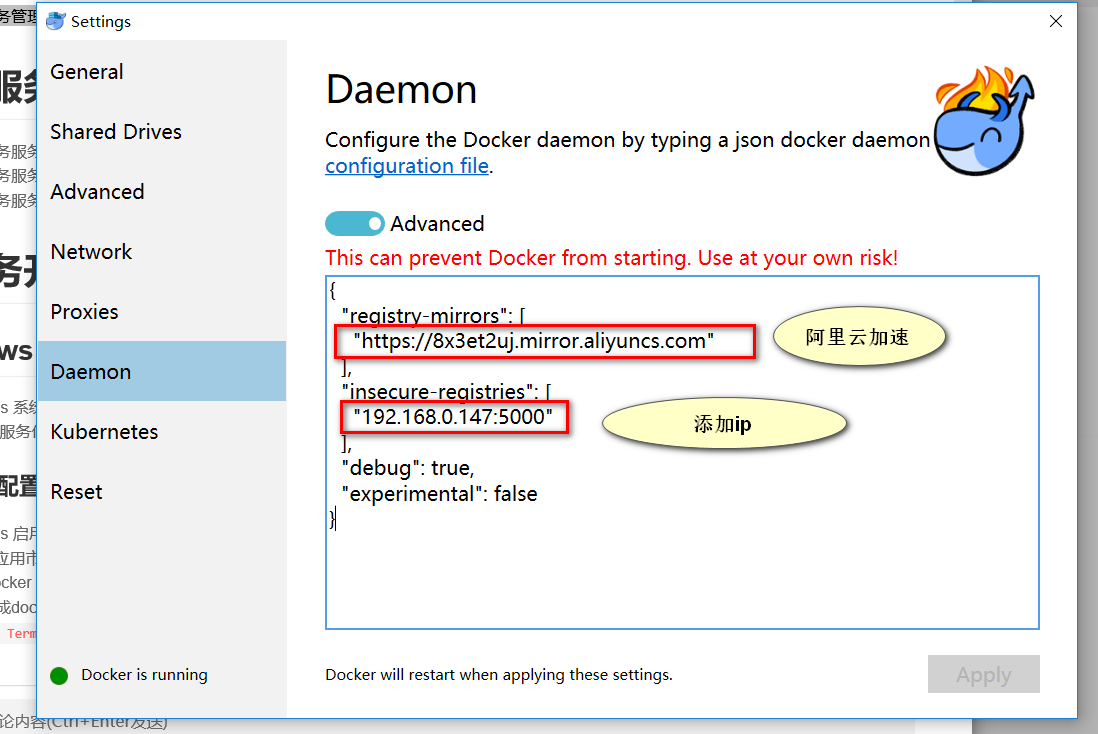
## 执行docker镜像，查看镜像服务是否正常

docker run -d -p 10008:10008 iqiyi

**第一个10008为本地端口，第二个10008为暴露的端口，iqiyi为镜像名**

## 如果推送到远程服务器上，可以这样

### 设置本地docker



### 推送到远程

#### 添加本地镜像到远程映射

docker tag iqiyi:latest 192.168.0.147:5000/iqiyi:latest

#### 推送到远程

docker push 192.168.0.147:5000/iqiyi:latest

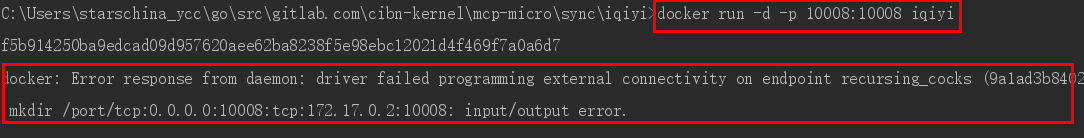
#### 查看远程镜像

curl 192.168.0.147:5000/v2/\_catalog

# **docker常见问题**

## docker run image后报错：

docker: Error response from daemon: driver failed programming external connectivity on endpoint recursing\_cocks



原因：docker服务启动时定义的自定义链DOCKER由于某种原因被清掉，重启docker服务及可重新生成自定义链DOCKER