第 3 章课程辅助资料

为了便于大家更好地理解课程中的知识，学院教研组配套整理了课程所需的辅助资料，供大家参考使用。

1. Docker image 云端下载与使用

```
# 启动 Docker

# 登入 Dockerhub
https://hub.docker.com/

# 寻找官方 image
 - 沒有前綴詞
 - name: hello-world image

docker images
docker pull hello-world
docker images
docker container run hello-world

# 进入讲师个人页面
https://hub.docker.com/repositories/uopsdod

# 寻找个人 image
 - 有账号前缀词 (ex. uopsdod/)
 - go to Repository page
 - 了解 tag 概念
 -- 不指定则为 :latest

docker pull uopsdod/deepblue-course-personal-image:v1
docker images
docker container run uopsdod/deepblue-course-personal-image:v1

docker pull uopsdod/deepblue-course-personal-image:v2
docker images
docker container run uopsdod/deepblue-course-personal-image:v2

docker pull uopsdod/deepblue-course-personal-image:latest
docker images
docker container run uopsdod/deepblue-course-personal-image:latest

# 使用 image id
```

```
docker images
docker container run [image id]

# 完成!
```

2. Docker 本地建立与使用

```
# 启动 Docker

# 建立 Dockerfile
 - 不用了解内部内容，专注在 image 的使用上即可
 - Dockerfile 命名需要一模一样

touch Dockerfile
=====
FROM alpine:3.7
RUN touch say.txt
RUN echo "You have built your first image from a Dockerfile. Great!" > say.txt
ENTRYPOINT ["cat", "./say.txt"]
=====

# 建立 image
docker build .
docker images
- 可以看到名称为 <none>

# 建立 image + 指定名称
docker build -t deepblue-course-first-image .
docker images
docker container run deepblue-course-first-image

# 建立 Dockerfile + build 参数使用

=====
FROM alpine:3.7
ARG my_name_is=nobody
RUN echo "You have built a new image from a Dockerfile. Well done, $my_name_is!" > say.txt
ENTRYPOINT ["cat", "./say.txt"]
=====

docker build -t deepblue-course-who-are-you-image .
docker images
docker container run deepblue-course-who-are-you-image
docker build -t deepblue-course-who-are-you-image --build-arg my_name_is="Ye Wenjie" .
docker images
docker container run deepblue-course-who-are-you-image

# 完成!
```

3. Docker 云端上传

```
# 启动 Docker

# 登录 dockerhub
docker logout
docker login

# 上传 image (失败范例)
docker images
docker push deepblue-course-who-are-you-image
- expected error: "denied: requested access to the resource is denied"

# 重新建立 image
- 加上账号前缀 (ex. uopsdod/)

docker build -t uopsdod/deepblue-course-who-are-you-image --build-arg my_name_is="Ye Wenjie" .
docker images
docker push uopsdod/deepblue-course-who-are-you-image

# 查看网页结果
https://hub.docker.com/repositories/uopsdod

# 完成!
```

4. Docker Container 实践示范

```
# 启动 Docker

# 使用 container
- 类别：瞬间关闭类别

docker container ls
docker pull alpine:3.7
docker images
docker container run alpine:3.7 echo "hey001"
docker container run alpine:3.7 ls /
docker run alpine:3.7 ls /
docker container ls
- expected: 空的

# 使用 container
- 类别: 用到主动离开为止

docker run alpine:3.7 /bin/sh
- expected: 瞬间关掉

docker run -it alpine:3.7 /bin/sh
/ # echo "hey001"
/ # ls
/ # exit

docker container ls
- expected: 还是空的

# 使用 container
- 类别: 永久在背景跑

docker run -d -it alpine:3.7 /bin/sh
docker container ls

docker run -d -it alpine:3.7 tail -f /dev/null
docker container ls

docker run -d -it alpine:3.7 ls /
docker container ls

docker container ls

# 使用 container + port mapping
docker pull nginx:1.23
docker images
docker run -d -p 8081:80 --name nginx001 nginx:1.23
docker container ls

http://localhost:8081

# 进入 container
docker exec -it c002 /bin/sh
/ # echo "hey001"
/ # ls
/ # exit


# 完成!

=====
```

5. Docker Container & image 本地清理

```
# 启动 Docker

# 关闭并清理 container - by name
docker container ls
docker container stop [contianer_id]
docker container ls
docker container ls -a
docker container rm [contianer_id]
docker container ls -a

# 删除 image
docker images
docker rmi nginx:1.23
docker images

# 重新启动
docker pull nginx:1.23
docker images

docker run -d -p 8081:80 --name nginx001 nginx:1.23
docker container ls

# 删除 image (失败范例)
docker rmi nginx:1.23
- expected error:
Error response from daemon: conflict: unable to remove repository reference "uopsdod/deepblue-course-who-are-you-image" (must force) - container ________ is using its referenced image ________

# 先删除 container (失败范例)
docker rm nginx001
- expected error:
You cannot remove a running container 785d320ea1c374fc048599067d380239a4378b77fe93914a13bb8102dcba05b3. Stop the
```

```
container before attempting removal or force remove

# 先刪除 container
docker container stop nginx001
docker container rm nginx001
docker container ls -a
docker rmi nginx:1.23
docker images

# 全部清理 - Container
docker container ls -a
docker container ls -a -q
docker stop $(docker container ls -a -q)
docker rm $(docker container ls -a -q)
docker container ls -a

# 全部清理 - Image (失败范例)
docker images
docker images -q
docker rmi $(docker images -q)
- expected error:

# 全部清理 - Image
docker rmi -f $(docker images -q)
docker images

# 完成!

=====
```