



**Hewlett Packard
Enterprise**

HPE University —— Docker专业课程





**Hewlett Packard
Enterprise**

Docker培训1



如果用Docker的话



你只需要5分钟！ 一条命令！！！！

```
docker run -it --name mysqlsrv1 -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123456  
mysql:5.7
```

```
docker exec -it mysqlsrv1 bash  
mysql -uroot -p123456  
show databases;
```

新开窗口执行

查看Docker版本与环境信息

docker version

```
Client:
Version:      20.10.6
API version:  1.41
Go version:   go1.13.15
Git commit:   370c289
Built:        Fri Apr  9 22:42:10 2021
OS/Arch:      linux/amd64
Context:      default
Experimental: true

Server: Docker Engine - Community
Engine:
Version:      20.10.6
API version:  1.41 (minimum version 1.12)
.....
```

docker info

```
Client:
Context:      default
Debug Mode: false

Server:
Containers: 10
Running: 1
Paused: 0
Stopped: 9
Images: 16
Server Version: 20.10.6
Storage Driver: overlay2
Backing Filesystem: xfs
Supports d_type: true
Native Overlay Diff: true
.....
```

Docker hello word

docker run hello-world

Unable to find image 'hello-world:latest' locally

latest: Pulling from library/hello-world

78445dd45222: Pull complete

Digest: sha256:c5515758d4c5e1e838e9cd307f6c6a0d620b5e07e6f927b07d05f6d12a1ac8d7

Status: Downloaded newer image for hello-world:latest

Hello from Docker!

This message shows that your installation appears to be working correctly.

运行一个nginx server容器

```
docker run -d -p 80:80 --name webserver nginx
```

```
Unable to find image 'nginx:latest' locally
```

```
latest: Pulling from library/nginx
```

```
ff3d52d8f55f: Pull complete
```

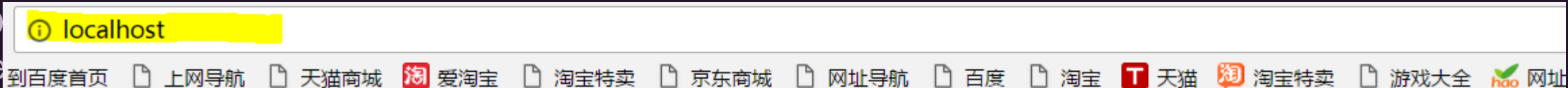
```
226f4ec56ba3: Pull complete
```

```
53d7dd52b97d: Pull complete
```

```
Digest: sha256:41ad9967ea448d7c2b203c699b429abe1ed5af331cd92533900c6d77490e0268
```

```
Status: Downloaded image
```

```
3f13352e
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Docker命令: docker run (上)

运行一个Docker容器，是最常用也最复杂的Docker命令

```
docker run -d -p 80:80 --name webserver nginx
```

放到后台执行

容器的名称，可以用这个名称
操作容器，比如删除容器

Image:tag
镜像（及版本）

哪个端口是主机的映射端口？ Try it out!!

Docker命令: docker run (中)

docker run -it busybox sh

启动容器并进入容器交互
(类似SSH)

执行镜像里的某个命令

```
[root@localhost ~]# docker run -it busybox sh
/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:AC:11:00:02
          inet addr:172.17.0.2  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe11:2/64 Scope:Link
```

命令的参数args

[root@localhost ~]# docker run busybox iplink show eth0

```
24: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff
```


Docker命令: docker run (下)

```
docker run -it -v /root:/mydata busybox sh
```

-v hostdir:vmdir, 产生了一个容器的Volume
将宿主机上的某个目录映射到容器里的某个目录

```
[root@localhost ~]# docker run -it -v /root:/mydata busybox sh
```

```
/ # ls /mydata/  
anaconda-ks.cfg mysql.tar  
/ # mkdir /mydata/test  
/ # ls /mydata/test  
/ # echo > /mydata/test/1.txt  
/ # ls /mydata/test/  
1.txt
```

提示: 主机上可以看到容器创建的文件, 反之亦然

Docker命令: docker exec
后台方式启动一个容器

docker run -d -p 80:80 --name webserver nginx

查看当前运行中的容器（加参数-a为查看所有）

docker ps

```
[root@localhost ~]# docker ps
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES |
|--------------|-------|-------------------------|---------------|--------------|--------------------|-----------|
| eb575c7d3581 | nginx | "nginx -g 'daemon ...'" | 3 minutes ago | Up 3 minutes | 0.0.0.0:80->80/tcp | webserver |

进入容器

容器的ID或者名称 执行容器中的某个命令

docker exec -it eb575c7d3581 bash

排查容器的练习

用busybox启动一个容器，验证nginx的容器正常工作

docker exec命令行方式进入nginx容器，获取nginx容器的IP地址（ip a）
Busybox启动一个容器，**并且ping nginx容器的IP**，确认可以网络互通
在busybox容器里用下列命令（bin目录下）行排查nginx容器是否正常：

- 1 wget <http://nginxip:80>
- 2 curl <http://nginxip:80>
- 3 telnet nginxip 80 或者8080，对比效果

Telnet可以测试任何TCP Server的端口是否可以连接，大家可以继续试试这种方式测试MySQL容器

Docker命令: docker inspect & docker log

可以用于Trouble shooting

查看一个容器的详细信息

```
docker inspect containerId
```

```
docker inspect eb575c7d3581|grep IPAddress  
  "SecondaryIPAddresses": null,  
  "IPAddress": "172.17.0.2",  
    "IPAddress": "172.17.0.2",
```

查看一个容器的输出日志（控制台日志）

```
docker logs containerid
```


容器传递环境变量

环境变量传递参数，可以多个-e来传递多个环境变量

```
docker run -it --name mysqlsrv1 -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123456 mysql:5.7
```

MySQL容器下面的3个环境变量参数表示创建一个特定的用户以及Database，并授权访问

- MYSQL_DATABASE
- MYSQL_USER
- MYSQL_PASSWORD

Try it out

新开窗口执行，验证可以连接到MySQL服务器

```
docker exec -it mysqlsrv1 bash  
mysql -uroot -p123456  
show databases;
```

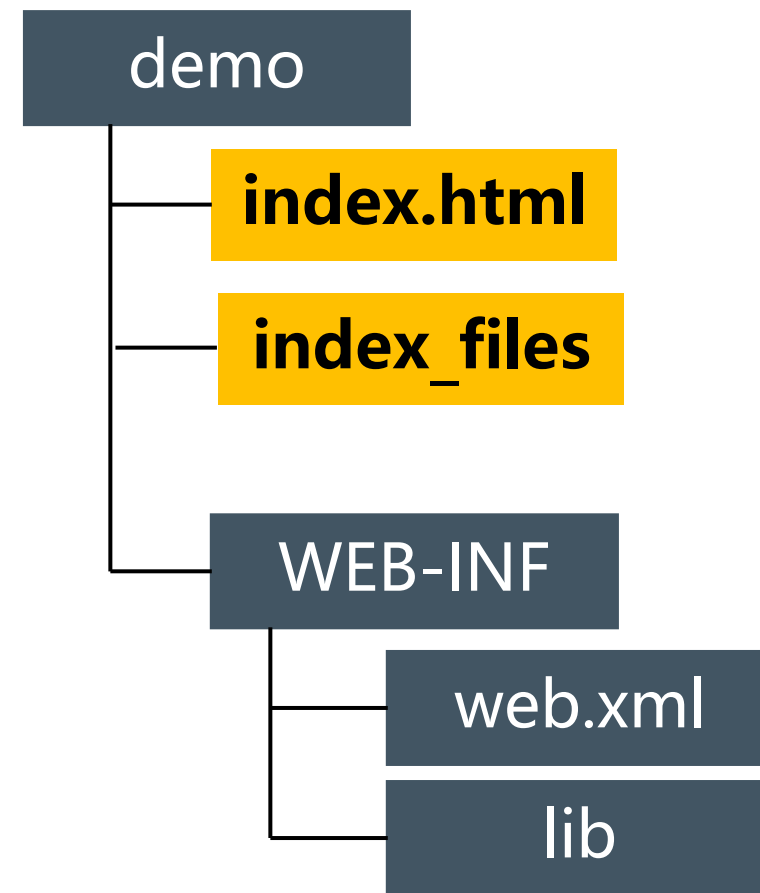
练习常用的Docker命令

docker run --help

| 命令 | 说明 |
|--------------------------|----------------|
| docker run | 运行一个容器 |
| docker ps -a | 查看所有容器，包括停止的容器 |
| docker inspect | 查看容器的详细信息 |
| docker start/stop | 启动或停止一个容器 |
| docker rm | 删除一个容器 |
| docker build | 镜像打包命令 |
| docker pull | 下载镜像 |
| docker images | 查看本地下载的镜像 |
| docker rmi | 删除某个镜像 |
| docker save xx -o xx.tar | 导出镜像 |
| docker load -o xx.tar | 导入镜像 |
| docker tag | 镜像打标签（版本） |
| docker push | 上传镜像到仓库 |

Build 镜像 (一)

```
FROM tomcat:9-alpine  
MAINTAINER bestme <bestme@hpe.com>  
ADD app /usr/local/tomcat/webapps/app
```



Build 镜像 (二)

/root/Dockerfile-demo

```
[root@docker_node Dockerfile-demo]# ./build-image.sh
```

```
build app image
```

```
Sending build context to Docker daemon 229.4 kB
```

```
Step 1/3 : FROM tomcat:9-alpine
```

```
---> 737982a24f9e
```

```
Step 2/3 : MAINTAINER bestme <bestme@hpe.com>
```

```
---> Running in 29b9ad656ca3
```

```
---> 5a0fd0d9de04
```

```
Removing intermediate container 29b9ad656ca3
```

```
Step 3/3 : ADD app /usr/local/tomcat/webapps/app
```

```
---> 196ab219711f
```

```
Removing intermediate container 490b060e2ba5
```

```
Successfully built 196ab219711f
```

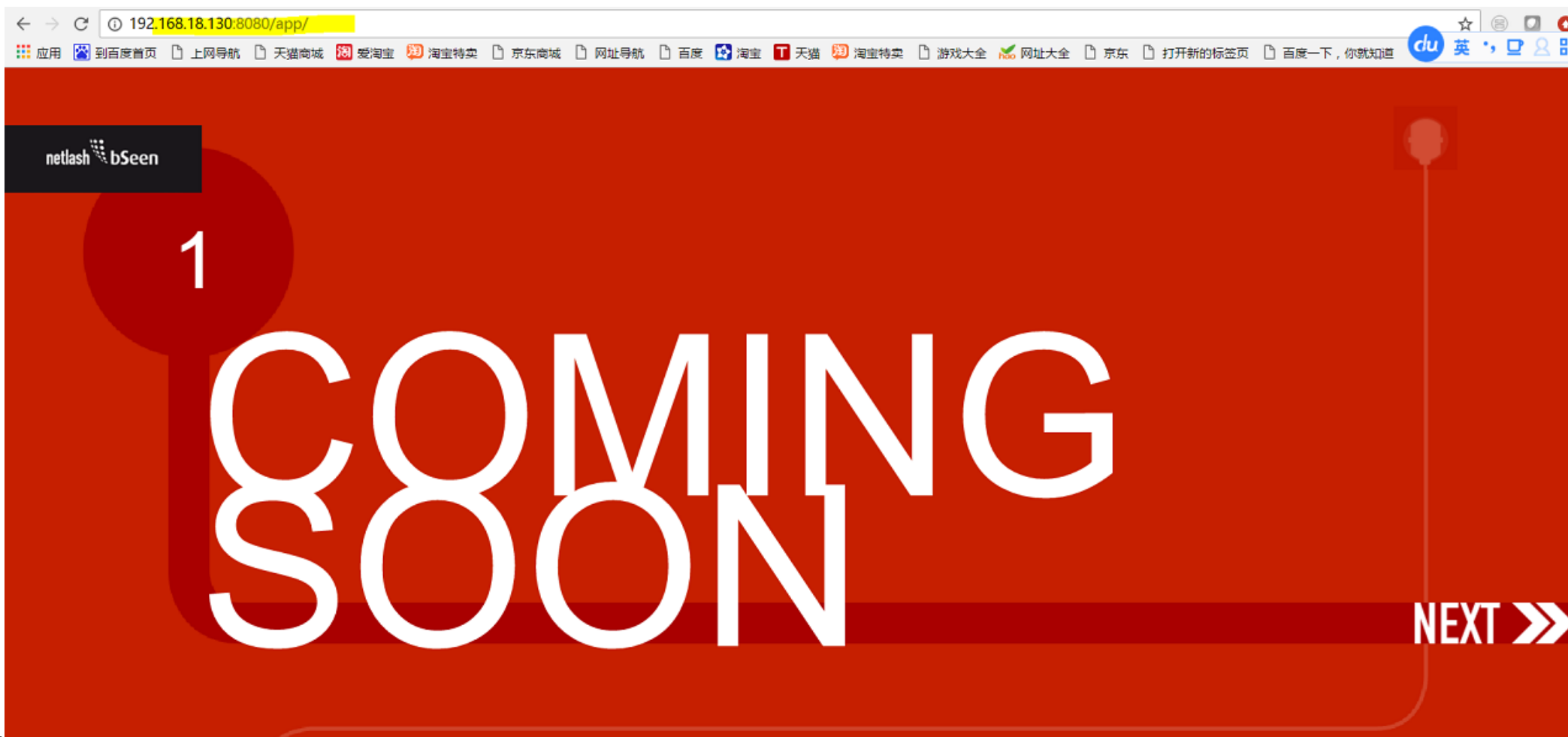
为什么用容器方式去构建镜像?
小组讨论

```
[root@docker_node Dockerfile-demo]# docker images
```

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|---------------|--------|--------------|--------------------|--------|
| hpe_app_image | latest | 196ab219711f | About a minute ago | 111 MB |

运行镜像

```
docker run -it -p 8080:8080 hpe_app_image
```



团队练习

- 修改index.html，列出本组所有人名字
- 重新打包生成镜像
- 传递到同组另外一个人的机器上
- 启动应用，验证结果

好看+速度，哪家最强？



Hewlett Packard
Enterprise

HPE University

地址：北京市 朝阳区 广顺南大街8号院 1号楼 利星行中心A座

邮箱：hpeu@hpe.com

