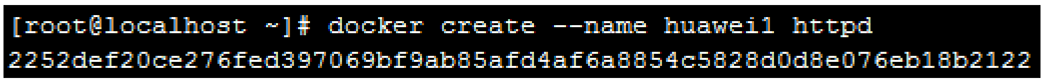
1.1 容器基础操作

1.1.1 运行第一个容器

步骤 1 创建一个名为 huawei1 的 httpd 容器。

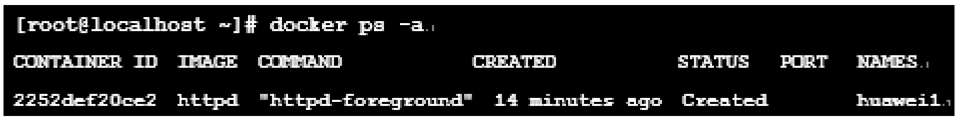
```
[root@localhost ~]# docker create --name huawei1 httpd
```



```
[root@localhost ~]# docker create --name huawei1 httpd
2252def20ce276fed397069bf9ab85afd4af6a8854c5828d0d8e076eb18b2122
```

步骤 2 查看该容器信息。

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



```
[root@localhost ~]# docker ps -a
CONTAINER ID  IMAGE  COMMAND             CREATED         STATUS    PORT   NAMES
2252def20ce2  httpd  "httpd-foreground"  14 minutes ago  Created          huawei1
```

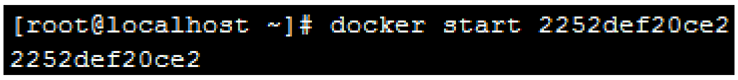
查询到容器 ID 为 2252def20ce2，容器名称为 huawei1，容器状态为已创建 Created。

步骤 3 启动容器 huawei1。

```
[root@localhost ~]# docker start huawei1
```

或者

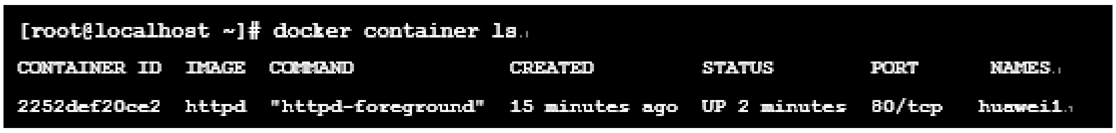
```
[root@localhost ~]# docker start 2252def20ce2
```



```
[root@localhost ~]# docker start 2252def20ce2
2252def20ce2
```

步骤 4 再次查看容器的 huawei1 信息，状态为 UP。

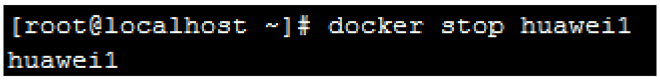
```
[root@localhost ~]# docker container ls
```



```
[root@localhost ~]# docker container ls
CONTAINER ID  IMAGE  COMMAND             CREATED         STATUS        PORT    NAMES
2252def20ce2  httpd  "httpd-foreground"  15 minutes ago  UP 2 minutes  80/tcp  huawei1
```

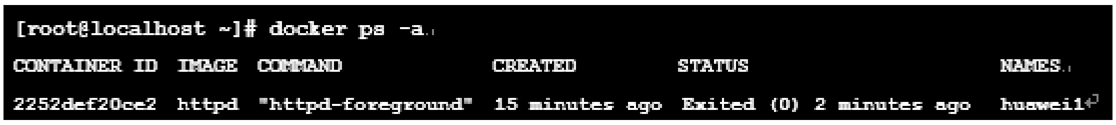
步骤 5 停止容器 huawei1，并查看到该容器状态为 Exited。

```
[root@localhost ~]# docker stop huawei1
```



```
[root@localhost ~]# docker stop huawei1
huawei1
```

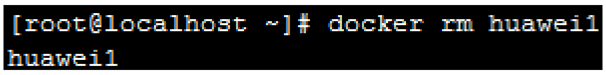
```
[root@localhost ~]# docker ps -a
```



```
[root@localhost ~]# docker ps -a
CONTAINER ID  IMAGE  COMMAND             CREATED         STATUS                     NAMES
2252def20ce2  httpd  "httpd-foreground"  15 minutes ago  Exited (0) 2 minutes ago   huawei1
```

步骤 6 删除容器 huawei1。

```
[root@localhost ~]# docker rm huawei1
```



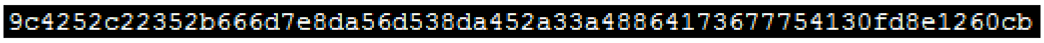
```
[root@localhost ~]# docker rm huawei1
huawei1
```

1.1.2 docker attach 进入容器

步骤 1 长期运行一个 centos 容器。

```
[root@localhost ~]# docker run -d centos /bin/bash -c "while true; do sleep 1;
echo Huawei; done"
```

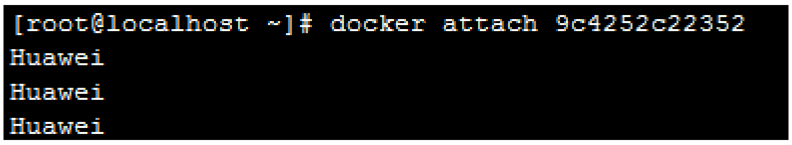
回显：



```
9c4252c22352b666d7e8da56d538da452a33a4886417367775413 0fd8e1260cb
```

步骤 2 使用 docker attach 命令直接进入这个 centos 容器终端。查看到每隔 1 秒钟输出字符“Huawei”。

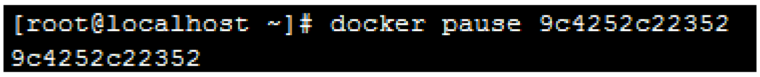
```
[root@localhost ~]# docker attach 9c4252c22352
```



```
[root@localhost ~]# docker attach 9c4252c22352
Huawei
Huawei
Huawei
```

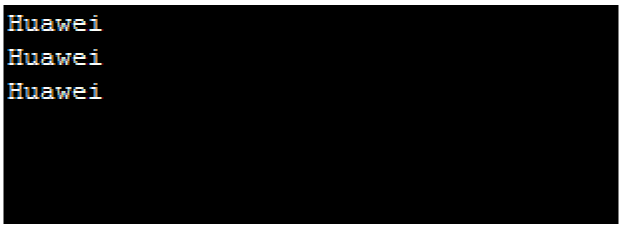
步骤 3 打开另一个宿主机终端，暂停这个容器。

```
[root@localhost ~]# docker pause 9c4252c22352
```



```
[root@localhost ~]# docker pause 9c4252c22352
9c4252c22352
```

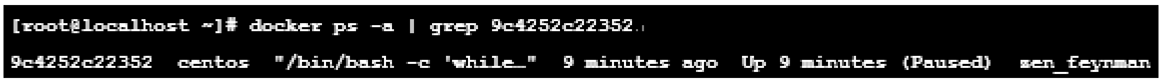
步骤 4 在 docker attach 进入容器的终端上，按“Enter”键，发现字符串“Huawei”已停止输出。



```
Huawei
Huawei
Huawei
```

步骤 5 查看该 centos 容器的状态，状态为 Paused。

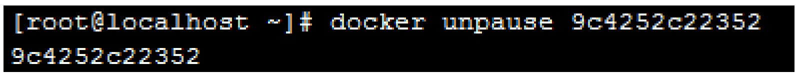
```
[root@localhost ~]# docker ps -a | grep 9c4252c22352
```



```
[root@localhost ~]# docker ps -a | grep 9c4252c22352
9c4252c22352   centos   "/bin/bash -c 'while…"   9 minutes ago   Up 9 minutes (Paused)   zen_feynman
```

步骤 6 恢复该 centos 容器的运行状态。

```
[root@localhost ~]# docker unpause 9c4252c22352
```

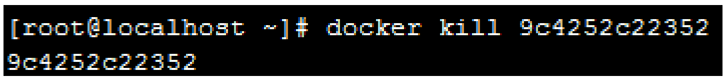


```
[root@localhost ~]# docker unpause 9c4252c22352
9c4252c22352
```

步骤 7 再次在 docker attach 进入容器的终端上查看，已恢复字符串“Huawei”的输出。

步骤 8 强制停止容器进程。

```
[root@localhost ~]# docker kill 9c4252c22352
```

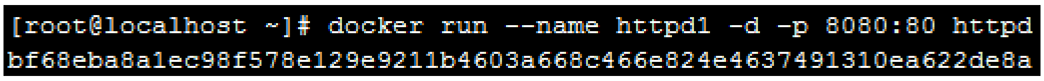


```
[root@localhost ~]# docker kill 9c4252c22352
9c4252c22352
```

1.1.3 docker exec 进入容器

步骤 1 在后台运行一个名为“httpd1”的 httpd 容器，并将其服务端口 80 映射到宿主机 8080 端口。

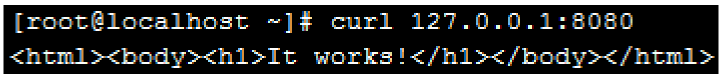
```
[root@localhost ~]# docker run --name httpd1 -d -p 8080:80 httpd
```



```
[root@localhost ~]# docker run --name httpd1 -d -p 8080:80 httpd
bf68eba8a1ec98f578e129e9211b4603a668c466e824e4637491310ea622de8a
```

步骤 2 访问容器 httpd1。

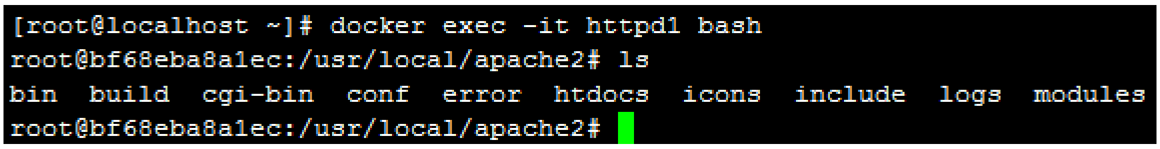
```
[root@localhost ~]# curl 127.0.0.1:8080
```



```
[root@localhost ~]# curl 127.0.0.1:8080
<html><body><h1>It works!</h1></body></html>
```

步骤 3 进入容器 httpd1。

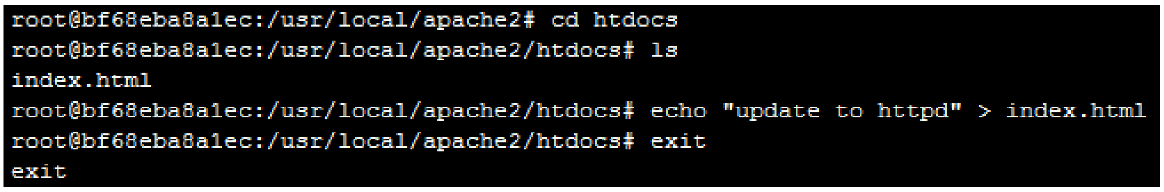
```
[root@localhost ~]# docker exec -it httpd1 bash
```



```
[root@localhost ~]# docker exec -it httpd1 bash
root@bf68eba8a1ec:/usr/local/apache2# ls
bin  build  cgi-bin  conf  error  htdocs  icons  include  logs  modules
root@bf68eba8a1ec:/usr/local/apache2# 
```

步骤 4 修改 httpd1 容器中静态文件内容，修改完成后输入“exit”退出。

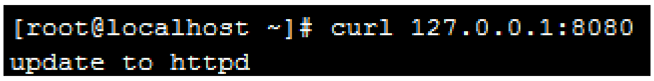
```
echo "update to httpd" > index.html
```



```
root@bf68eba8a1ec:/usr/local/apache2# cd htdocs
root@bf68eba8a1ec:/usr/local/apache2/htdocs# ls
index.html
root@bf68eba8a1ec:/usr/local/apache2/htdocs# echo "update to httpd" > index.html
root@bf68eba8a1ec:/usr/local/apache2/htdocs# exit
exit
```

步骤 5 再次访问容器 httpd1。容器依旧可正常访问，说明“exit”退出并不会导致容器进程结束。

```
[root@localhost ~]# curl 127.0.0.1:8080
```



```
[root@localhost ~]# curl 127.0.0.1:8080
update to httpd
```

步骤 6 为方便后续实验，将本小节中的容器删除。

```
docker kill
docker rm
```