

# 实验三 部署Docker环境

---

- 1. 连接实验服务器
- 2. 安装Docker服务
- 3. Docker基本操作

## 1. 连接实验服务器

一：创建资源

- 1. 在页面左侧，单击 云产品资源 下拉菜单，查看本次实验资源。
- 2. 单击 免费开通 创建所需资源。

资源创建过程需要1~3分钟。完成实验资源的创建后，您可以在 云产品资源 列表查看已创建的资源信息，例如：IP地址、用户名和密码等。

3.连接ECS服务器

Windows用户：使用putty远程登录ECS服务器，putty下载地址如下：

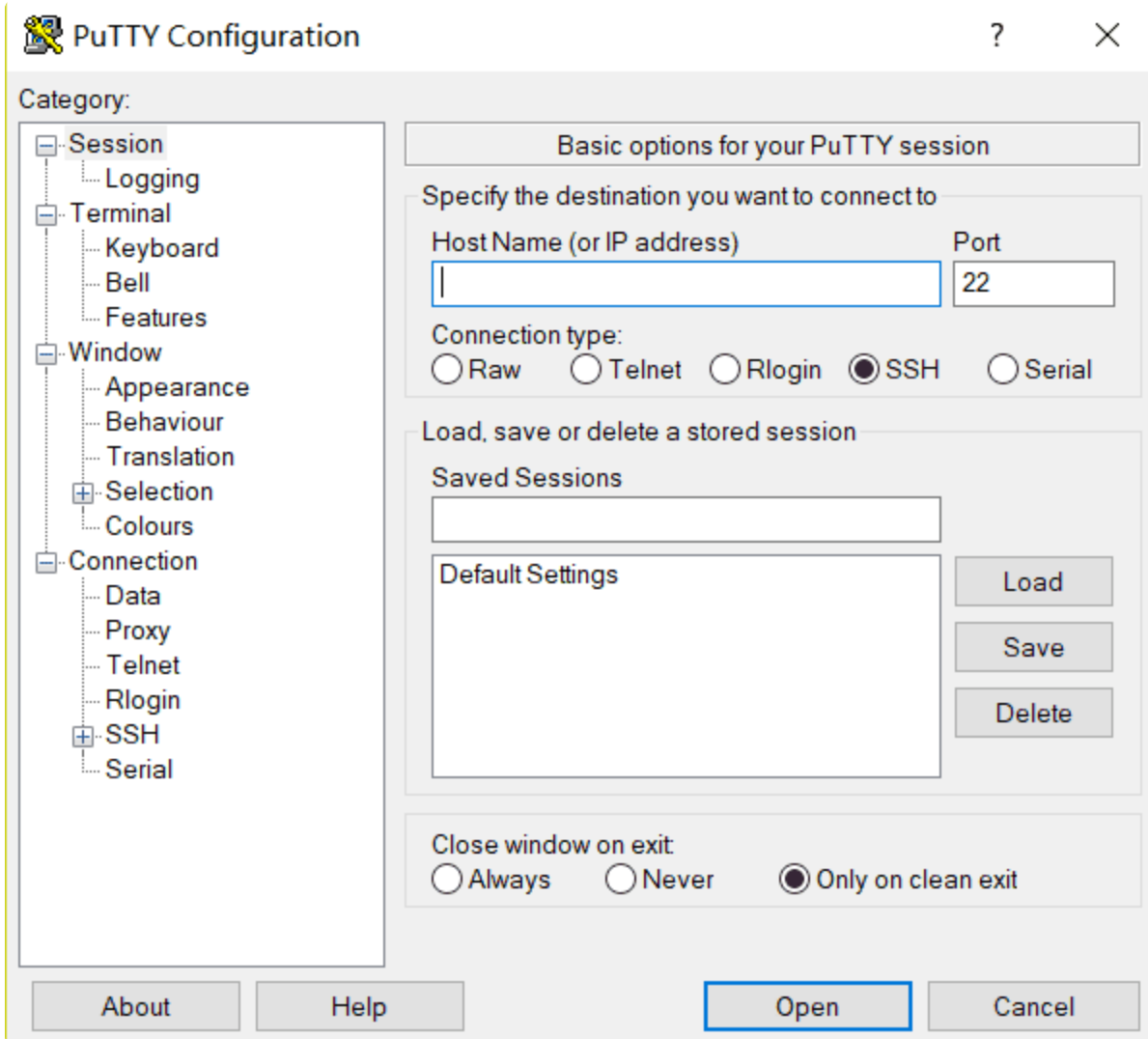
64-bit: <https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe>

32-bit: <https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe>

Mac用户：使用系统自带的终端“Terminal”连接ECS服务器。

4.远程登录到ECS服务器

双击打开putty软件，在“Host Name (or IP address)”中输入ECS公网地址。



完成后，点击“Open”。

在弹出的对话框中，输入ECS服务器的登录用户名root，输入ECS服务器的登录密码（此处密码不会显示）。

登录成功后显示如下信息：

```
login as: root
root@101.132.33.252's password:

Welcome to Alibaba Cloud Elastic Compute Service !

[root@iZuf6gi2ckefjvrjrjh5wdZ ~]#
```

## 2. 安装Docker服务

1. 运行以下命令安装Docker的依赖库。这里yum-utils 提供了 yum-config-manager, 并且 device mapper存储驱动程序需要 device-mapper-persistent-data 和 lvm2。

▼ Plain Text |

```
1 yum install -y yum-utils device-mapper-persistent-data lvm2
```

2. 运行以下命令添加Docker CE的软件源信息。Docker版本分为 CE (Community Edition: 社区版) 和 EE (Enterprise Edition: 企业版), 这里选择Docker CE。

▼ Plain Text |

```
1 yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
```

3. 运行以下命令安装Docker CE。

▼ Plain Text |

```
1 yum -y install docker-ce
```

4. 运行以下命令查看Docker版本:

▼ Plain Text |

```
1 docker -v
```

5. 运行以下命令启动Docker服务

▼ Plain Text |

```
1 systemctl start docker
```

6. 运行以下命令查看Docker服务状态

▼ Plain Text |

```
1 service docker status
```

Docker启动成功后显示如下信息:

```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2021-02-17 17:24:48 CST; 50s ago
     Docs: https://docs.docker.com
```

说明：停止Docker服务使用命令： `systemctl stop docker`

## 7. 运行以下命令测试运行Docker Hello World

```
1 docker run hello-world
```

```
Hello from Docker!  
This message shows that your installation appears to be working correctly.
```

## 3. Docker基本操作

本步骤在Docker容器中通过 Python Flask运行一个WEB应用

### 1. 列出镜像列表

```
1 docker images
```

命令运行结果如下所示：

```
[root@iZuf630dygwd6zxpo86zy7Z ~]# docker images  
REPOSITORY      TAG           IMAGE ID       CREATED        SIZE  
hello-world      latest        bf756fb1ae65   13 months ago 13.3kB
```

**说明：**REPOSITORY表示镜像的仓库源。同一仓库源可以有多个TAG，代表这个仓库源的不同版本。如Ubuntu仓库源里，有15.10、14.04等多个不同的版本，使用 REPOSITORY:TAG 来定义不同的镜像。

### 2.拉取镜像：把指定镜像拉取到本地

```
1 docker pull training/webapp
```

命令运行结果如下所示：

```
[root@iZuf630dygwd6zxpo86zy7Z ~]# docker pull training/webapp
Using default tag: latest
latest: Pulling from training/webapp
Image docker.io/training/webapp:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
e190868d63f8: Pull complete
909cd34c6fd7: Pull complete
0b9bfabab7c1: Pull complete
a3ed95caeb02: Pull complete
10bbbc0fc0ff: Pull complete
fca59b508e9f: Pull complete
e7ae2541b15b: Pull complete
9dd97ef58ce9: Pull complete
a4c1b0cb7af7: Pull complete
Digest: sha256:06e9c1983bd6d5db5fba376ccd63bfa529e8d02f23d5079b8f74a616308fb11d
Status: Downloaded newer image for training/webapp:latest
docker.io/training/webapp:latest
```

说明：拉取镜像需要较长时间，当拉取镜像失败时，建议配置DockerHub镜像加速器。具体方法是增加一个文件/etc/docker/daemon.json，并添加上registry-mirrors键值，步骤如下：

```
1 vi /etc/docker/daemon.json
```

在文件中添加以下内容

```
1 "registry-mirrors": ["https://registry.docker-cn.com"]
```

完成后点击“ECS”键，然后输入:wq保存退出文档。再重新执行拉取镜像命令。

### 3.运行容器

```
1 docker run -d -P training/webapp python app.py
```

参数说明：-d:让容器在后台运行

-P:将容器内部使用的网络端口映射到主机上

命令运行结果如下所示：

```
[root@iZuf630dygwd6zxpo86zy7Z ~]# docker run -d -P training/webapp python app.py
88bd74928ed78d4e19d29f3d1641d5f7a17230814eeaff4c557f7b00c5cdae04
```

### 4.查看正在运行的容器

```
1 docker ps
```

命令运行结果如下所示：

```
[root@iZuf630dygwd6zxpo86zy7Z ~]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
88bd74928ed7   training/webapp "python app.py"         About a minute ago Up About
a minute      0.0.0.0:49153->5000/tcp wizardly_feistel
```

可以看到 Docker 开放了 5000 端口（默认 Python Flask 端口）映射到主机端口 49153 上

**说明：**docker ps -a 列出所有容器（包括未运行的）。

5. 使用 ECS 实例 IP 和端口号，通过浏览器访问 WEB 应用。

在浏览器地址栏输入 http://ECS 实例公网地址:端口号 并按回车键，本示例运行结果如下：

A screenshot of a web browser window. The address bar shows the URL 'http://106.14.185.212:49153/'. Below the address bar, the text 'Hello world!' is displayed in a blue, monospace-style font.

6. 查看 WEB 应用程序日志，命令格式：docker logs 容器ID 或者名字

```
1 docker logs -f 88bd74928ed7
```

参数说明：-f：让 docker logs 像使用 tail -f 一样来输出容器内部的标准输出。

本示例运行结果如下：

```
[root@iZuf630dygwd6zxpo86zy7Z ~]# docker logs -f 88bd74928ed7
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
222.188.157.10 - - [17/Feb/2021 09:47:47] "GET /favicon.ico HTTP/1.1" 404 -
222.188.157.10 - - [17/Feb/2021 09:47:47] "GET /favicon.ico HTTP/1.1" 404 -
222.188.157.10 - - [17/Feb/2021 09:47:48] "GET /favicon.ico HTTP/1.1" 404 -
222.188.157.10 - - [17/Feb/2021 09:47:48] "GET /favicon.ico HTTP/1.1" 404 -
```

7. 停止 WEB 应用容器，命令格式：docker stop 容器ID 或者名字

```
docker stop 88bd74928ed7
```

本示例运行结果如下：

```
[root@iZuf630dygwd6zxpo86zy7Z ~]# docker stop 88bd74928ed7
88bd74928ed7
```

8. 移除 WEB 应用容器，命令格式：docker rm 容器ID 或者名字

▼		Plain Text
1	docker rm 88bd74928ed7	

**注意：**删除容器时，容器必须是停止状态，否则会报错误。