

常用的Docker命令和使用方法

[拉取（Pull）镜像](#)

[查看已下载的镜像](#)

[创建运行容器](#)

[列出正在运行的容器](#)

[停止和启动容器](#)

[进入容器](#)

[查看容器信息](#)

[查看容器日志](#)

[查看容器内文件系统的变化](#)

[删除容器和镜像](#)

[重命名容器](#)

[从旧容器复制数据到主机：](#)

[从主机复制数据到新容器：](#)

拉取（Pull）镜像

当我们在Docker中部署一个应用程序时，需要使用一个称为"镜像"的模板来创建容器。镜像包含了运行应用程序所需的所有文件系统、代码和依赖项。在Docker中，镜像系统指的是操作系统镜像（redhat, ubuntu）；

你可以通过**docker pull**命令来下载Docker镜像。例如，如果你想拉取Ubuntu的官方镜像，可以运行以下命令：



Plain Text |

```
1  docker pull ubuntu
```

查看已下载的镜像

使用**docker images**命令可以列出所有本地已下载的Docker镜像：



Plain Text |

```
1  docker images
```

创建运行容器

1、**docker run**：这是创建和运行容器最常见的方式。

```
1  docker run -it ubuntu
```

这个命令创建了一个基于Ubuntu镜像的容器，并在其中启动一个交互式bash终端。通过**-it**选项，它为容器分配了一个交互式终端，并允许用户与容器进行交互。

当你运行该命令后，Docker会创建一个基于Ubuntu镜像的容器，并在容器内部启动一个bash终端。你可以在终端中执行各种bash命令，就像在一个Ubuntu系统中一样。允许你与容器进行交互，并在终端中执行任意命令。

2、容器中运行**echo**命令并输出"Hello Docker!"。

```
1  docker run ubuntu /bin/echo "Hello Docker!"
```

这个命令是简单地运行一个容器来执行**/bin/echo "Hello Docker!"**命令，并输出"Hello Docker!"。然后，容器完成了任务并退出。这个命令在容器内部执行了一个命令，但没有创建交互式终端。

运行该命令后，Docker会创建一个基于Ubuntu镜像的容器，然后在容器内部执行**/bin/echo "Hello Docker!"**命令，并将"Hello Docker!"输出到标准输出。完成输出后，容器就会停止运行。

列出正在运行的容器

使用**docker ps**命令可以列出当前正在运行的容器：

```
1  docker ps
```

如果想查看所有包括已停止的容器，可以加上**-a**选项：

```
1  docker ps -a
```

停止和启动容器

使用**docker stop**命令可以停止一个正在运行的容器：

▼ Plain Text |

```
1 docker stop <容器ID或名称>
```

使用**docker start**命令可以启动一个已经停止的容器：

▼ Plain Text |

```
1 docker start <容器ID或名称>
```

进入容器

有时候你可能需要进入容器的命令行环境来进行调试或执行其他任务。可以使用**docker exec**命令进入正在运行的容器：

上面的命令将进入容器的bash终端。

▼ Plain Text |

```
1 docker exec -it <容器ID或名称> bash
```

查看容器信息

docker inspect 命令用于查看 Docker 容器的详细信息，包括容器的配置、网络设置、卷挂载、环境变量、ip 等等。

▼ Plain Text |

```
1 docker inspect <容器ID或名称>
```

查看容器日志

查看容器的日志输出，可以使用容器的ID或名称。

▼ Plain Text |

```
1 docker logs <容器ID或名称>
```

查看容器内文件系统的变化

查看容器内文件系统的变化，它会显示与基本镜像相比哪些文件被添加、修改或删除。

▼ Verilog |

```
1 docker diff <容器ID或名称>
```

删除容器和镜像

使用**docker rm**命令可以删除一个已经停止的容器：

▼ Plain Text |

```
1 docker rm <容器ID或名称>
```

使用**docker rmi**命令可以删除一个本地的Docker镜像：

在进行删除操作之前，请确保你已经停止了相应的容器

▼ Plain Text |

```
1 docker rmi <镜像ID或名称>
```

重命名容器

如果容器正在运行，首先停止它。使用 **docker rename** 命令来重命名容器。

▼ Plain Text |

```
1 docker rename <old_container_name> <new_container_name>
```

将 **<old_container_name>** 替换为当前的容器名称，将 **<new_container_name>** 替换为您希望的新容器名称

从旧容器复制数据到主机：

▼ Plain Text |

```
1 docker cp <old_container_name>:<container_path> <host_path>
```

<old_container_name>：旧容器的名称。

<container_path>：<容器ID或名称>。

<host_path>：目标主机上存储复制数据的路径。

从主机复制数据到新容器：



Plain Text |

```
1  docker cp <host_path> <new_container_name>:<container_path>
```

<host_path>： 之前从旧容器复制到主机的数据所在路径。

<new_container_name>： 新容器的名称。

<container_path>： <容器ID或名称>。