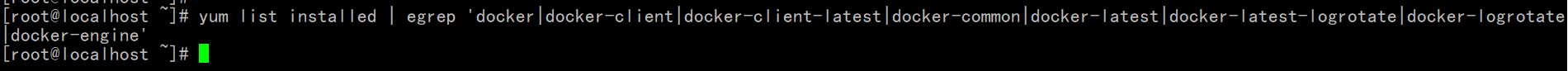
**任务一 安装Docker**

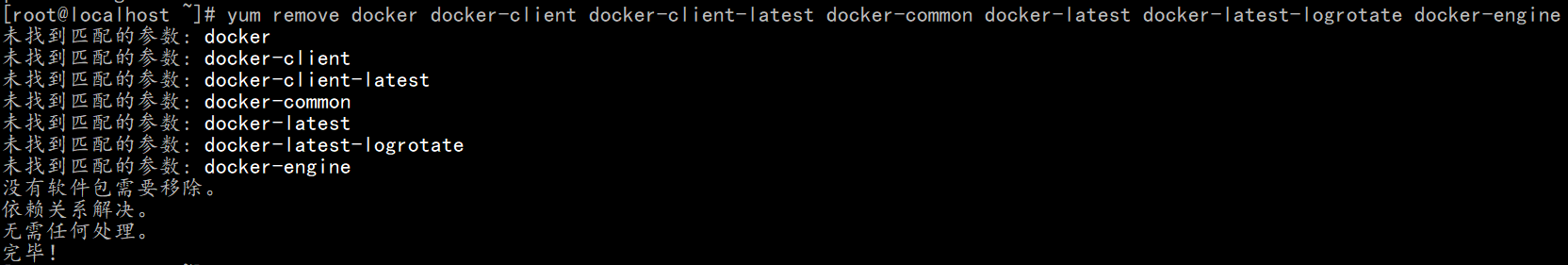
1. **查看系统内核版本**

****

1. **查询系统中是否安装有Docker旧版本软件**

****

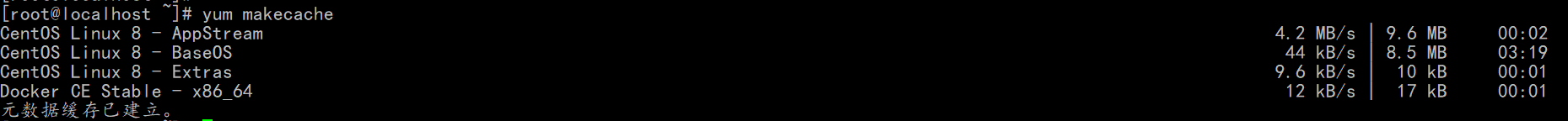
1. **删除旧版软件**

****

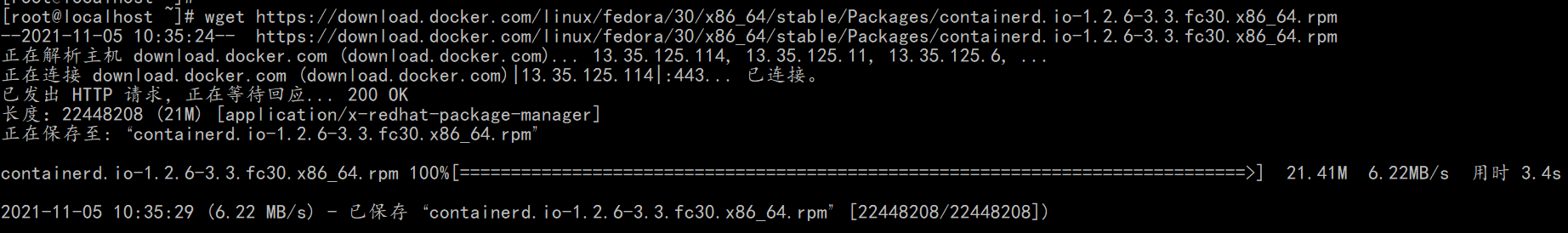
1. **获取仓库配置文件**

****

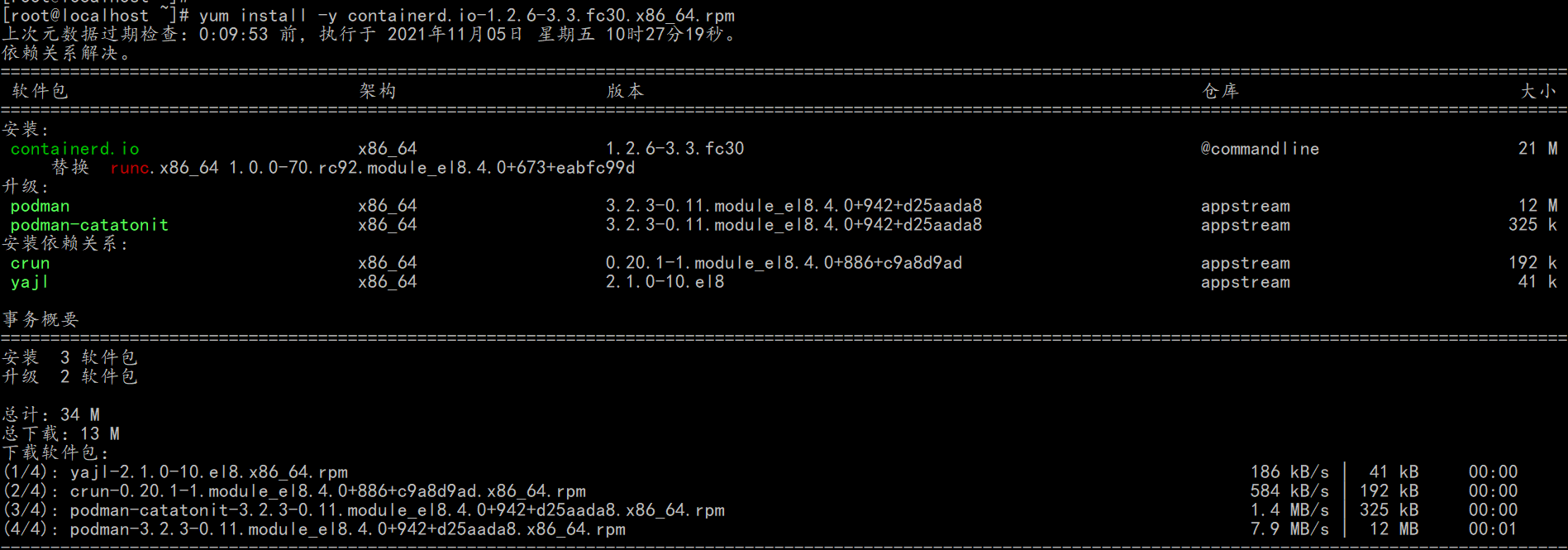
1. **使配置文件生效**

****

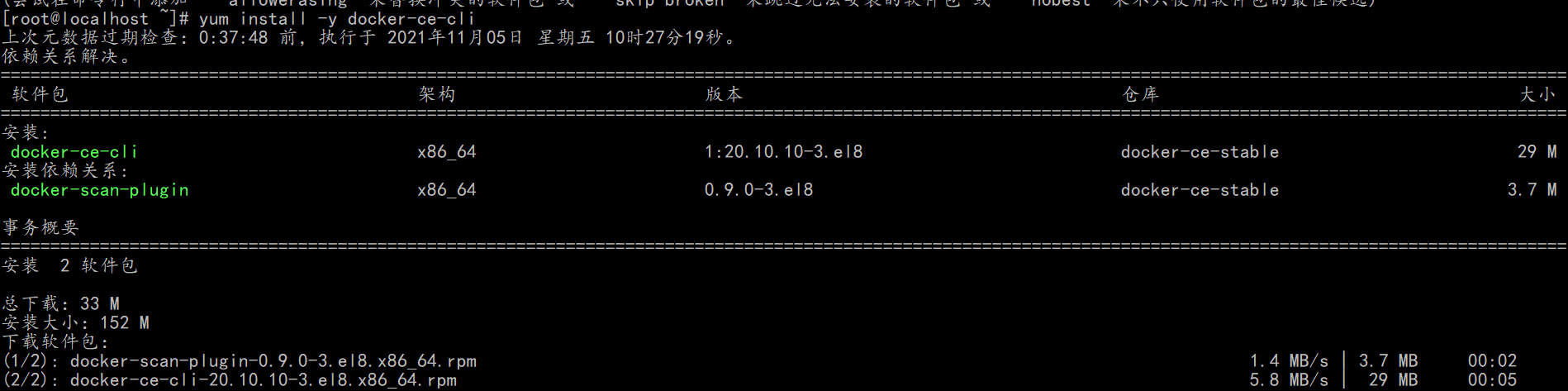
1. **在线安装Docker软件**

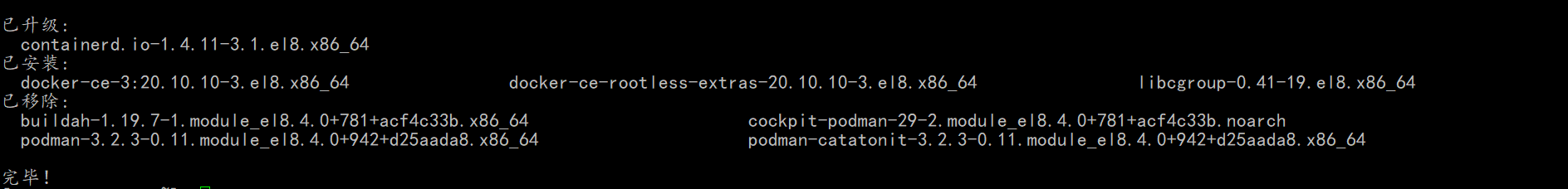
****

1. **安装containerd.io-1.2.6-3.3.fc30.x86\_64.rpm**

****

1. **安装Docker软件**

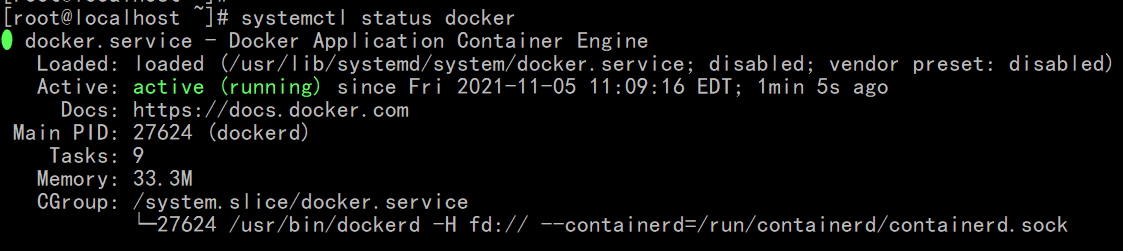
****

****

1. **启动Docker服务**

****

1. **查看Docker运行信息**

****

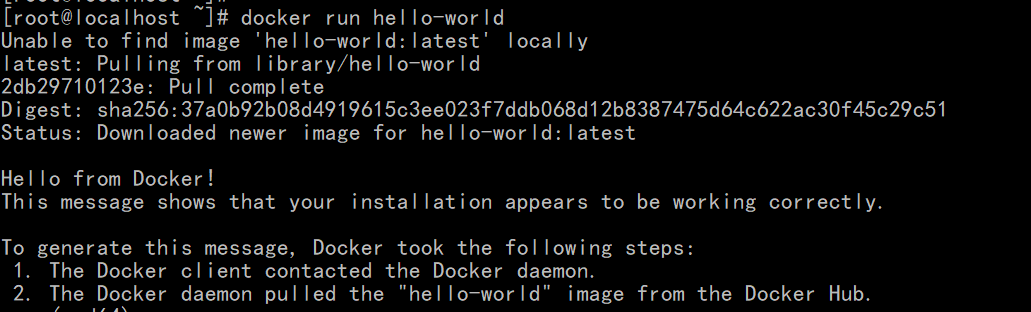
1. **设置Docker服务开机自启动**

****

1. **确认是否设置为开机自启动**

****

1. **运行hello-world容器，验证Docker是否安装成功**

****

**任务二 使用Docker创建新的镜像**

1. **关闭防火墙**

****

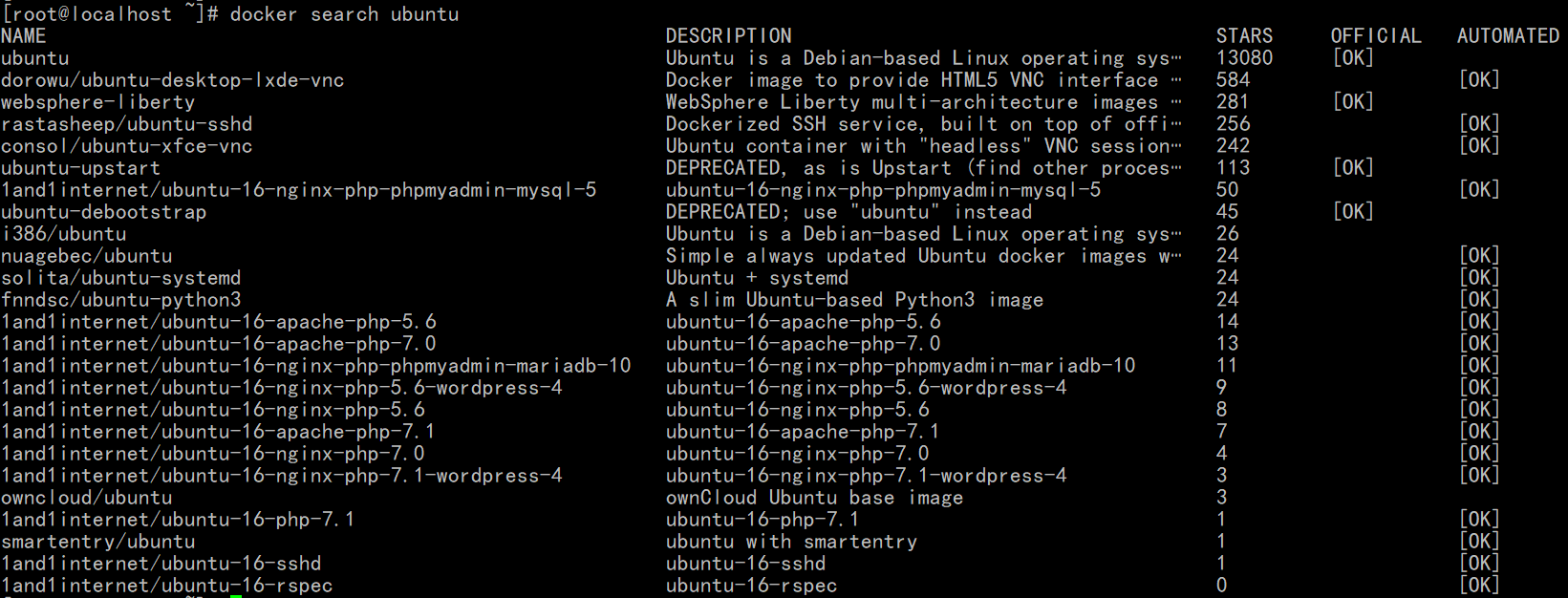
1. **将SELinux设置为permissive模式**

****

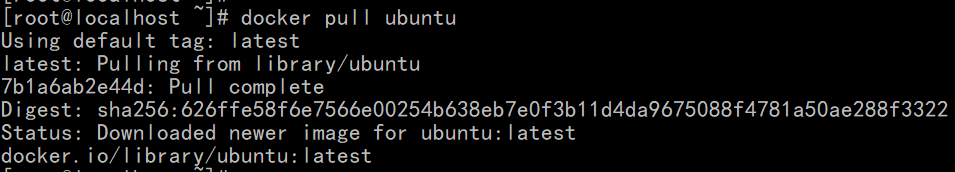
1. **重启Docker服务**

****

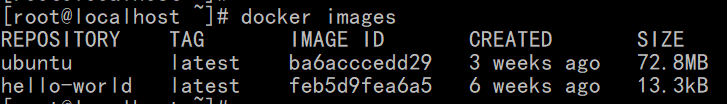
1. **检索DockerHub仓库中的ubuntu镜像**

****

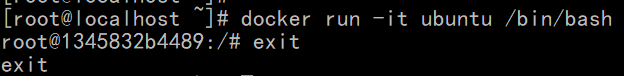
1. **下载ubuntu镜像**

****

1. **列出本地所有镜像**

****

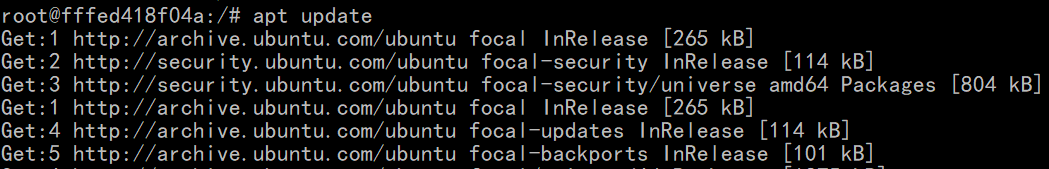
1. **启动ubuntu镜像中的Docker容器**

****

1. **启动ubuntu容器**

****

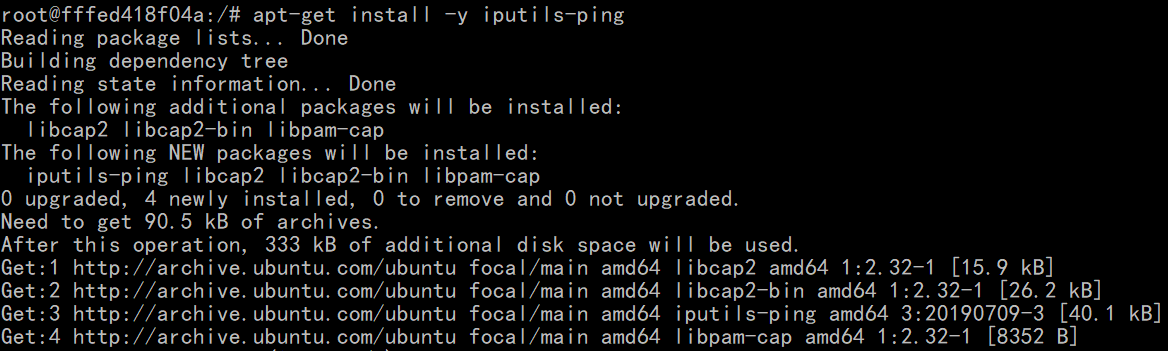
1. **对ubuntu容器检查更新**

****

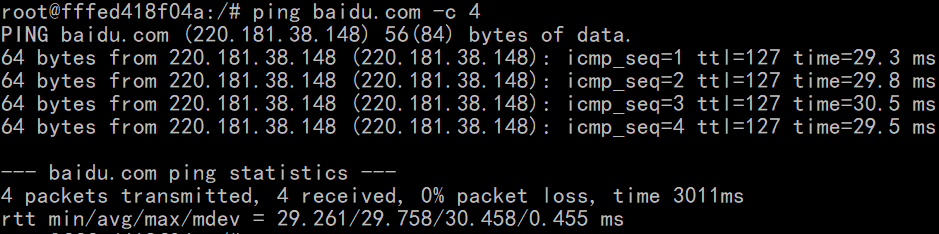
1. **测试Docker容器网络连通性**

****

1. **安装iputils-ping**

****

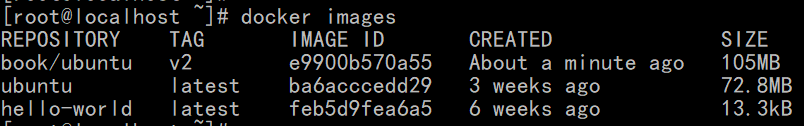
1. **测试Docker容器网络连通性**

****

1. **创建新的Docker容器镜像**

****

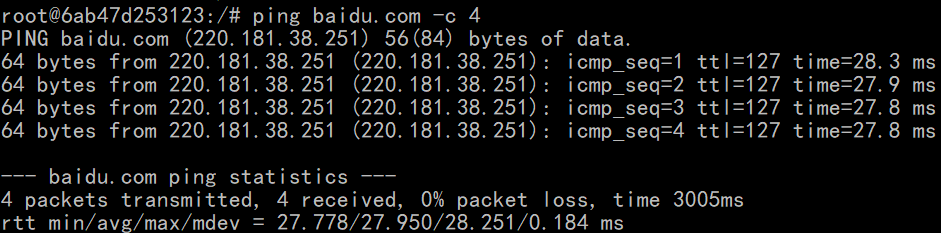
1. **查看创建成功的Docker镜像**

****

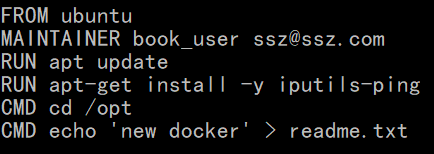
1. **使用新创建的Docker镜像运行容器**

****

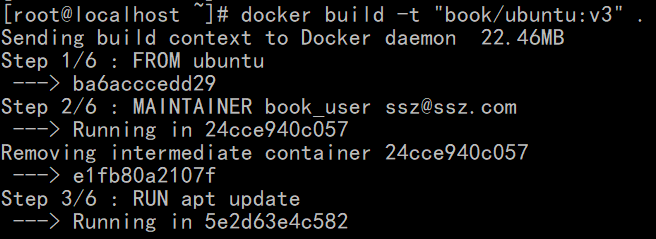
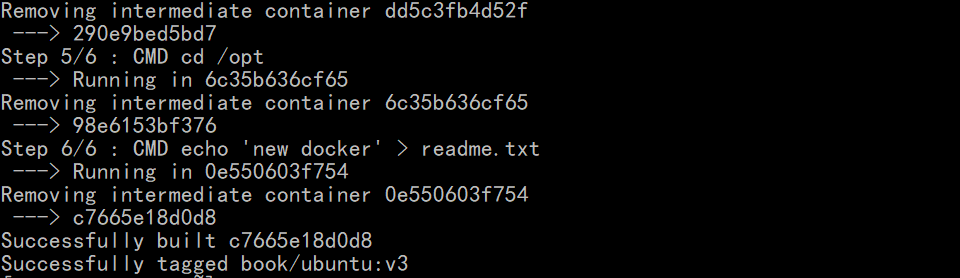
1. **测试网络连通性**

****

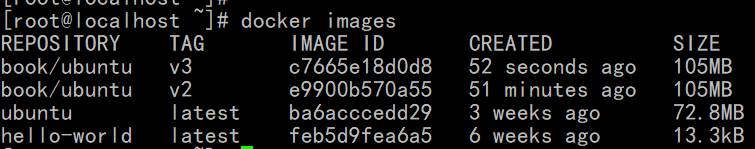
1. **配置Dockerfile文件**

****

1. **创建Docker镜像**

****

1. **查看创建的Docker镜像**

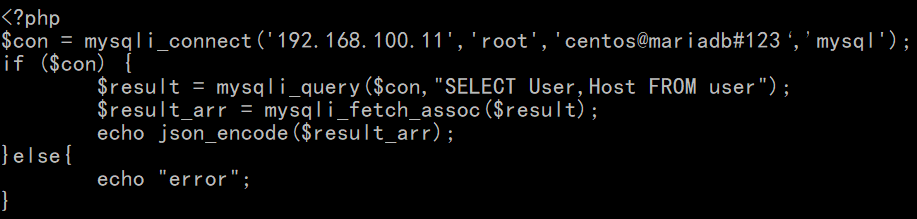
****

1. **删除hello-world镜像**

****

**任务三 使用Docker发布PHP程序**

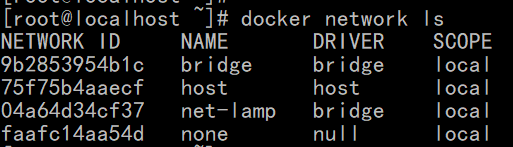
1. **配置PHP文件**

****

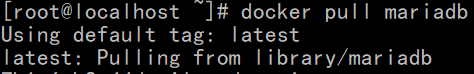
1. **创建Docker网络**

****

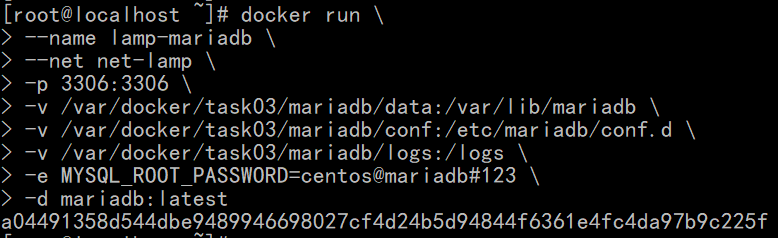
1. **查看已创建的Docker网络**

****

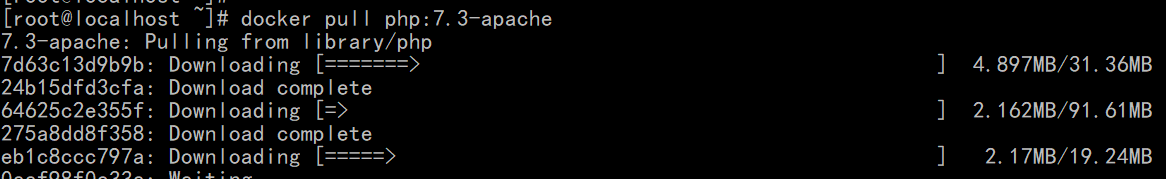
1. **下载mariadb镜像**

****

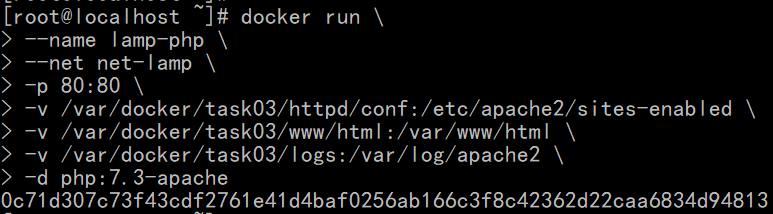
1. **启动mariadb容器**

****

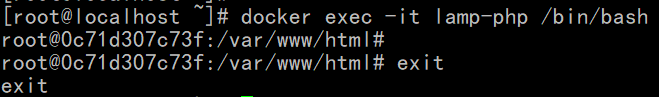
1. **下载php:7.3-apache镜像**

****

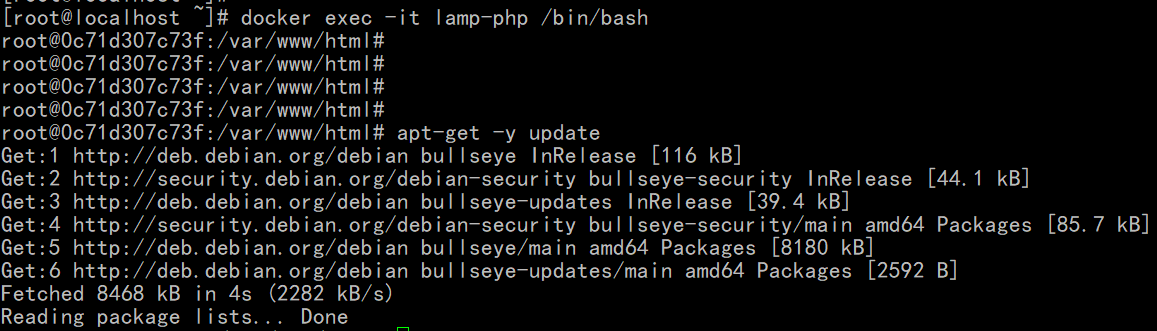
1. **启动PHP容器**

****

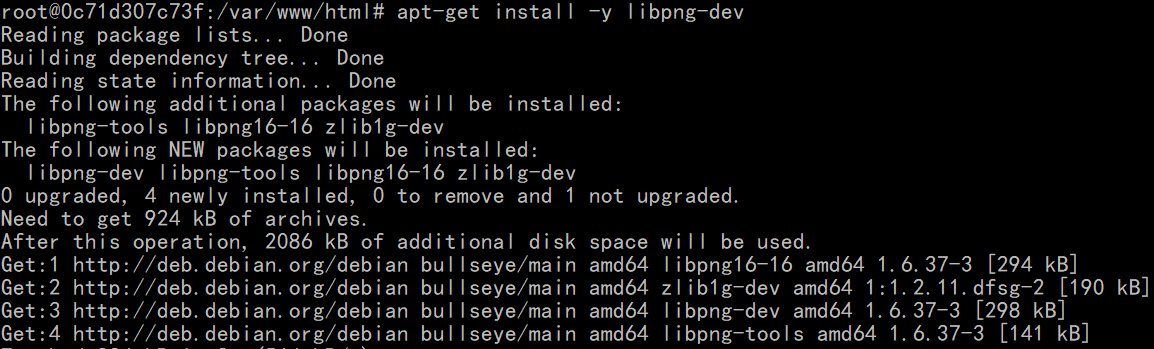
1. **链接Docker容器**

****

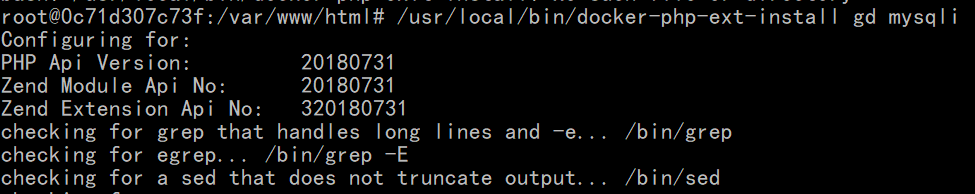
1. **对Docker容器进行升级**

****

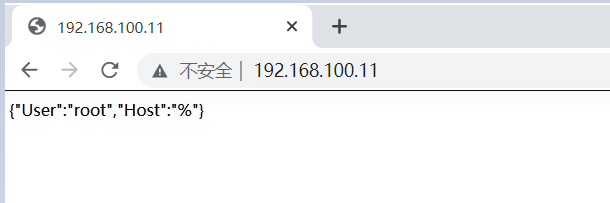
1. **安装编译依赖包**

****

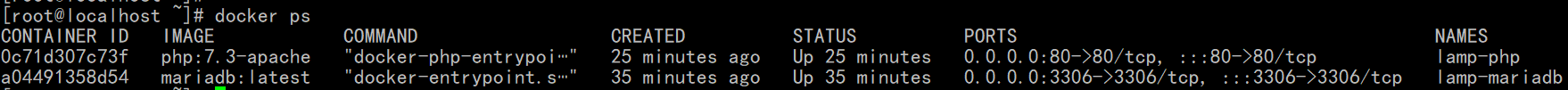
1. **安装mysqli模块**

****

1. **验证LAMP是否部署成功**

****

1. **查看运行中的Docker容器**

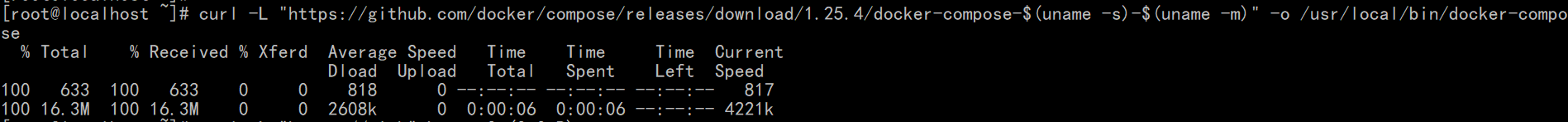
****

1. **查看容器资源消耗情况**

****

**任务四 使用Docker Compose发布业务**

1. **安装Docker Compose**

****

1. **设置可执行权限**

****

1. **测试docker-compose是否安装成功**

****

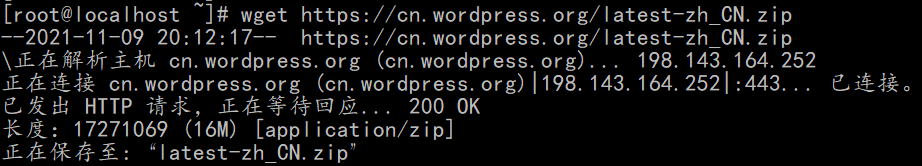
1. **创建docker-compose工作目录**

****

1. **准备WordPress安装程序**

****

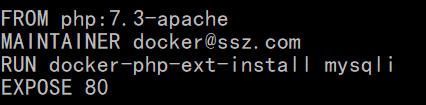
1. **下载WordPress**

****

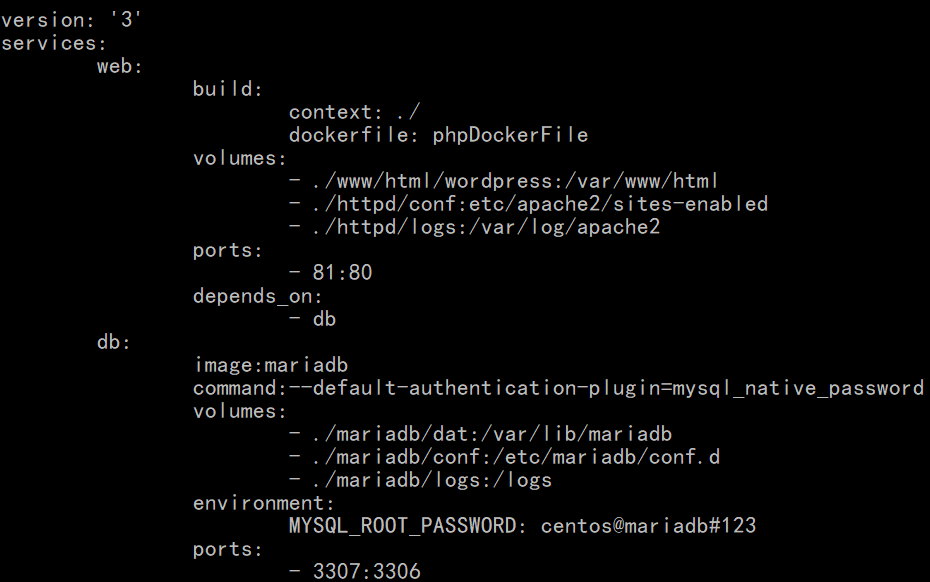
1. **解压latest-zh\_CN.zip**

****

1. **定义php容器**

****

1. **定义程序运行环境**

****

1. **验证WordPress是否部署成功**

****