无锡城市职业技术学院

**实 验（实 训）报 告**

**所属课程名称**  云平台构建与运维

**班 级** 计网2201

**学 号**  1422040316

**姓 名**  杨恩赐

**指导教师**  徐峰

无锡城市职业技术学院教务处监制

**项目名称：Docker容器的安装与使用** 日期：2024.5.7

|  |
| --- |
| **一、实验（实训）概述：** |
| **【目的及要求】**  **学会Docker容器的安装与使用**  **尝试使用Kunectl运行容器**  **【实验环境】**  **所需镜像：CentOS-7-x86\_64-Everything-2009.iso**  **所需平台：VMware Workstation**  **所需文件：K8S.tar.gz** |
| **二、实验（实训）内容：** |
| **【实验（实训）过程】（步骤、记录、数据、程序等）** 配置hosts    关闭防火墙和SELinux    关闭Swap    配置YUM  * **首先通过WinSCP将所需要的镜像和k8s.tar.gz上传到master节点下的/root/目录**        * **创建挂载目录**      * **编辑fstab**      * **挂载镜像**      * **解压缩Kubernetes**      * **编辑YUM配置文件**          * **安装vsftpd并进行编辑**        * **启动和启用服务**     **Node节点进行配置YUM文件**     配置NTP **master节点**  **我们需要做安装chrony并编辑配置文件，启动服务查看时间源。（与node节点更改配置文件有所区别）**        **node节点：**  **我们需要做安装chrony并编辑配置文件，启动服务查看时间源。（与master节点更改配置文件有所区别）**     配置路由转发    配置IPVS  * **编辑文件**        * **增加执行权限**      * **执行脚本**      * **显示已经载入的模块**      * **安装ipset软件包**     **综上所述：**   安装Docker  * **安装yum-utils**      * **安装支持的软件**      * **安装Docker**      * **创建文件目录**      * **编辑Docker配置文件**      * **启用Docker配置文件**      * **查看docker info**    安装Kubeadm工具 **安装软件并启动服务**      **启动服务** 初始化Kubernetes集群  * **加载镜像**      * **初始化Kubernetes集群**        * **配置kubectl工具**        * **查看集群状态**    配置Kubernetes网络  将node节点加入集群 **将K8S.tar.gz传到node节点中**    **解压缩Kubernetes**    **在nade节点上输入master节点初始化Kubernetes集群的时候弹出来的Token提示**      **在master节点下查看各个节点的状态**   安装Dashboard     **至此，原生Kubernetes云平台部署完成**  **使用Kubectl运行容器** 下载镜像  使用kubectl运行容器  * **创建deployment**      * **查看Pod是否正常运行**      * **产看所有deployment**      * **采用NodePort的方式暴露nginx服务**      * **查看Service**      * **物理机使用浏览器访问nginx应用**      * **Pod动态伸缩**      * **产看Pod数量**      * **删除deployment、 service**     **【结论】（结果、分析）**  通过该实验，我初步接触到容器这个概念，并在做实验的过程中去慢慢感受他的强大。在这里简单阐述一下一个容器的运行过程。  **准备镜像**：首先，根据用户定义的 Dockerfile 或者其他容器配置文件，Docker 引擎会从镜像仓库中拉取相应的镜像。镜像包含了运行容器所需的文件系统、环境变量、依赖项等信息。  **创建容器**：一旦镜像被拉取到本地，Docker 引擎会根据镜像创建一个容器实例。这个实例是镜像的运行时表示，它拥有自己的文件系统、网络配置、进程空间等。  **启动容器**：一旦容器创建完成，Docker 引擎会启动容器，并且根据容器配置执行相应的启动命令。这可能涉及到运行特定的应用程序、启动服务等操作。  运行应用：一旦容器启动，其中定义的应用程序或服务开始运行。这些应用程序可以访问容器内的文件系统、网络、环境变量等资源，并在容器内部执行相应的操作。  **监控和管理**：Docker 引擎会持续监控容器的运行状态，包括资源利用率、网络连接等。同时，用户也可以通过 Docker 命令或者管理工具来管理和操作容器，如启动、停止、删除等。  结束容器：当应用程序执行完成或者用户手动停止容器时，容器会被终止。在容器终止时，Docker 引擎会清理相应的资源，包括文件系统、网络配置等，并将容器状态标记为已停止。 |
| **三、指导教师评语及成绩：** |
| **评语：**  **成绩： 指导教师签名：徐峰**  **批阅日期：2023年 月 日** |