**山东大学 计算机科学与技术 学院**

**云计算技术 课程实验报告**

|  |  |  |  |
| --- | --- | --- | --- |
| **学号：**201900130133 | **姓名：**施政良 | | **班级：**四班 |
| **实验题目：**面向PaaS的CloudFoundry练习 | | | |
| **实验学时：**2 | | **实验日期：**2020-04-03 | |
| **实验目的：**了解CloudFoundry编程环境的部署和配置，完成实验环境及实验工具的熟悉，撰写实验报告。  **具体包括：**了解CloudFoundry编程环境的部署和配置，完成实验环境及实验工具的熟悉，撰写实验报告。 | | | |
| **硬件环境：**  联网计算机一台 | | | |
| **软件环境：**  Windows or Linux（ubuntu18.04） | | | |
| **实验步骤与内容：**  **实验步骤概述：**  本次实验主要涉及有关Cloud Foundry环境的配置，并了解其基本概念和应用。具体实验内容如下所示：   1. 了解Cloud Foundry的基本概念，包括基本应用以及系统架构等 2. 完成编程环境的部署和配置、并熟悉基本工具。 3. 对比Cloud Foundry和其他云平台的特点   为了方便起见，在本次实验中使用wsl ubuntu18.04版本作为实验主机，具体实验内容如下所示.  **具体实验内容**   1. Cloud Foundry 介绍   Cloud Foundry是业界第一个开源PaaS云平台，它支持多种框架、语言、运行时环境、云平台及应用服务，使开发人员能够在几秒钟内进行应用程序的部署和扩展，无需担心任何基础架构的问题。   1. Colud Foundry主要特性   从初始开发到所有测试阶段再到部署，Cloud Foundry支持应用程序开发的完整生命周期，因此作为持续交付的解决方案而广受推崇。Cloud Foundry基于容器的架构支持各类云服务供应商，同时支持以任何编程语言运行的应用程序。这一支持多个云部署的环境允许开发人员可利用适合特定应用程序工作负载的云平台，根据需要在短短几分钟内对这些工作负载进行迁移，而无需更改应用程序。了典型技术栈的层次，并将传统 IT 模型与云平台模型进行了比较：     1. Cloud Foundry与其他平台的兼容性   Cloud Foundry平台可从Cloud Foundry Foundation作为开源软件获取，也可从众多商业服务供应商处作为软件产品或软件服务获取。Cloud Foundry是一项开源软件，因此可供任何人使用。部署Cloud Foundry涉及使用由Cloud Foundry基金会管理的另一项开源工具——Cloud Foundry BOSH部署系统与底层基础架构进行交互。   1. **具体的配置过程** 2. 注册账户：首先安装文档中的说明验证邮箱并注册，具体过程如下：     注册完成之后显示如下页面。     1. 下载命令行工具cf cli：安装官方文档中的提示通过命令行进行下载。   在终端输入以下命令，   |  | | --- | | wget -q -O - https://packages.cloudfoundry.org/debian/cli.cloudfoundry.org.key | sudo apt-key add -  echo "deb https://packages.cloudfoundry.org/debian stable main" | sudo tee /etc/apt/sources.list.d/cloudfoundry-cli.list |       之后更新sudo,输入以下命令   |  | | --- | | sudo apt-get update |   运行结果如下所示：    之后安装命令行工具   |  | | --- | | sudo apt-get install cf7-cli |     最后打印帮助文档测试是否安装成功：结果如下所示：    通过上述打印信息可知，帮助文档打印成功，说明命令行工具安装正确。   1. 设置环境并登录：首先设置API，如下图所示     之后使用短期验证码登录     1. 具体测试进行测试：   （1）使用Cloud Foundry CLI 安装 Cloud Foundry 本地插件  在终端输入：   |  | | --- | | cf install-plugin cflocal |     安装成功后有如下提示    （2）以hello wold项目为例进行部署，在终端中使用git命令进行克隆    下载完成后进入子文件夹并更改文件   |  | | --- | | cf push cf-helloworld |  |  | | --- | | python-3.8.1 |   之后删除原文件   |  | | --- | | cf delete py-helloworld |   重新运行push命令即cf push cf-helloworld，运行成功，查看控制台有以下结果 | | | |
| **结论分析与体会：**  **结论分析：**   1. Cloud Foundry与其他云平台的比较   分析：目前市面上有多重流形的云平台，在本实验中以Kubernetes为例将Cloud Foundry和 Kubernetes进行对比。  （1）相同之处：   1. 两者都使用容器的思想来隔离应用和系统其它组件。 2. 两者都可以既运行在公有云（AWS、Azure、GCP）上，也可以运行在预置型云平台，如使用Vmware vsphere的云平台上。 3. 两者都提供了混合环境上的运行能力，允许你在不同的云平台运行应用以提高可用性，甚至支持应用在公有云和私有云上同时运行。 4. Pivotal Cloud Foundry的最新版本也开始支持Kubernetes作为通用的容器运行时   （2）不同之处：   1. Cloud Foundry是一个独立于云的平台即服务解决方案，属于一种开源项目，可通过各种私有云发行版和公有云实例获得；而Kubernetes是一个来源于谷歌Borg项目的开源云平台 2. Kubernetes首要的功能是一个容器运行时。通常是被用来运行Docker容器。有一些解决方案基于Kubernetes提供了PaaS体验，比如RedHat OpenShift。相比于Kuberetes，Cloud Foundry（CF）基于容器的架构，可运行任何编程语言的应用程序。使用现有工具将应用程序部署到CF，无需修改代码。 3. Cloud Foundry提供了云、开发者框架和应用服务的选择，可以更快、更容易的构建、测试、发布和大规模部署应用程序。相比之下Kubernetes部署和使用更加的复杂   **体会：**  通过本次实验，我初步了解了Cloud Foundry的基本应用。Cloud Foundry是由相对独立的多个模块构成的分布式系统，每个模块单独存在和运行，各模块之间通过消息机制和API接口进行通信。  目前Cloud Foundry云平台主要有Router、Cloud Controller、Health Manager、DEA、NFS、NATS、Cloud Controller Database以及Service等模块组成。所有的访问请求都通过Router进行转发，分别由云控制器Cloud Controller和应用运行代理DEA模块进行请求响应，应用生命周期管理Health Manager模块负责监控和管理整个应用在云平台上的正常运行，云平台的各种应用服务由Services模块提供，可以灵活扩展。  综上，本次实验中通过实际接触流行的云计算系统和平台使我对“云”的概念有了更加深刻的认识。 | | | |