

南昌大学实验报告

姓名：柳杰

学号：6103215019

邮箱地址：liu__jie@outlook.com

专业班级：网络工程151班

实验日期：2018年5月12日

课程名称：Linux程序设计

实验项目名称

- Build it and Share it

实验目的

- Understanding the mechanism of IaaS hierarchy
- Trying to launch certain engineering services
- Understanding the process of build a visible web service

实验基础

- openstack部署和基本操作
- docker基本原理和基础操作

实验步骤结果及数据

- packstack安装openstack基本环境

```
# 安装Centos7 mini版本操作系统
# 安装前准备
systemctl disable firewalld NetworkManager
systemctl enable network
#关闭selinux
#查看当前selinux状态
getenforce
getenforce 0
#安装一些组件
yum install vim net-tools wget -y
#修改selinux配置文件
vim /etc/selinux/config
SELINUX=disabled
# 网络配置
```

```
BOOTPROTO=none
IPADDR=192.168.1.###
PREFIX=24
GATEWAY=192.168.1.1
DNS1=222.204.2.20
# 语言环境
vim /etc/environment
# 写入下面内容
LANG=en_US.utf-8
LC_ALL=en_US.utf-8
# 安装软件源
yum install -y centos-release-openstack-queens
yum update -y
yum install -y openstack-packstack
# 生成应答文件，选择相应功能
packstack --gen-answer-file=answer.txt
# 开始安装
packstack --answer-file=answer.txt
```

- 配置虚拟机网络环境

创建网络



- 网络 *
- 子网
- 子网详情

名称

external

创建一个新的网络。额外地，网络中的子网可以在向导的下一步中创建。

项目 *

admin

供应商网络类型 * ?

Flat

物理网络 * ?

extnet

☒ 启用管理员状态

☐ 共享的

☐ 外部网络

☒ 创建子网

可用域提示 ?

nova

- 取消
- « 返回
- 下一步 »

创建网络



网络 *

子网

子网详情

子网名称

external

网络地址 ?

192.168.1.0/24

IP版本

IPv4

网关IP ?

192.168.1.1

☐ 禁用网关

创建关联到这个网络的子网。您必须输入有效的“网络地址”和“网关IP”。如果您不输入“网关IP”，将默认使用该网络的第一个IP地址。如果您不想使用网关，请勾选“禁用网关”复选框。点击“子网详情”标签可进行高级配置。

取消

« 返回

下一步 »

创建网络



网络 *

子网

子网详情

☒ 激活DHCP

为子网指定扩展属性

分配地址池 ?

192.168.1.200,192.168.1.220

DNS服务器 ?

222.204.2.20

主机路由 ?

取消

« 返回

已创建

- 创建虚拟机并安装docker

```
yum install docker -y
```

- 部署Homework2项目代码

拉取Go语言基本环境

```
Connected (unencrypted) to: QEMU (instance-00000019)
[root@localhost ~]# systemctl start docker
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
[root@localhost ~]# docker pull golang
Using default tag: latest
Trying to pull repository docker.io/library/golang ...
latest: Pulling from docker.io/library/golang
cc1a78bfd46b: Downloading [>] 1 456.7 kB/45.32 MB
6861473222a6: Downloading [====>] 1 1.178 MB/10.77 MB
7e0b9c3b5ae0: Downloading [=====>] 1 736.2 kB/4.336 MB
3ee98735f56f: Waiting
32ecd1fcfe1a: Waiting
9889d58a42e5: Waiting
337bad6698be: Waiting
```

环境测试

```
docker run -it -p 8080:8080 镜像ID
```

```
Connected (unencrypted) to: QEMU (instance-00000019)
root@3363e1a4d753:/go# go
Go is a tool for managing Go source code.

Usage:

    go command [arguments]

The commands are:

    build      compile packages and dependencies
    clean      remove object files and cached files
    doc        show documentation for package or symbol
    env        print Go environment information
    bug        start a bug report
    fix        update packages to use new APIs
    fmt        gofmt (reformat) package sources
    generate    generate Go files by processing source
    get        download and install packages and dependencies
    install    compile and install packages and dependencies
    list       list packages
    run        compile and run Go program
    test       test packages
    tool       run specified go tool
    version    print Go version
    vet        report likely mistakes in packages

Use "go help [command]" for more information about a command.

Additional help topics:

    c          calling between Go and C
    buildmode  build modes
    cache      build and test caching
    filetype   file types
    gopath     GOPATH environment variable
    environment environment variables
    importpath import path syntax
    packages   package lists
    testflag   testing flags
    testfunc   testing functions

Use "go help [topic]" for more information about that topic.

root@3363e1a4d753:/go# _
```

运行测试代码

```
root@3363e1a4d753:/go/HomeWork2# ls
api.go  diskusage  diskusage.go  login.gtpl  testdir
root@3363e1a4d753:/go/HomeWork2# gp run api.go
._api.go  _login.gtpl  api.go  diskusage
root@3363e1a4d753:/go/HomeWork2# gp run api.go
._api.go  _login.gtpl  api.go  diskusage
root@3363e1a4d753:/go/HomeWork2# gp run api.go
bash: gp: command not found
root@3363e1a4d753:/go/HomeWork2# go run api.go
method: GET
```

192.168.1.126:8080/homework2

192.168.1.126:8080/homework2

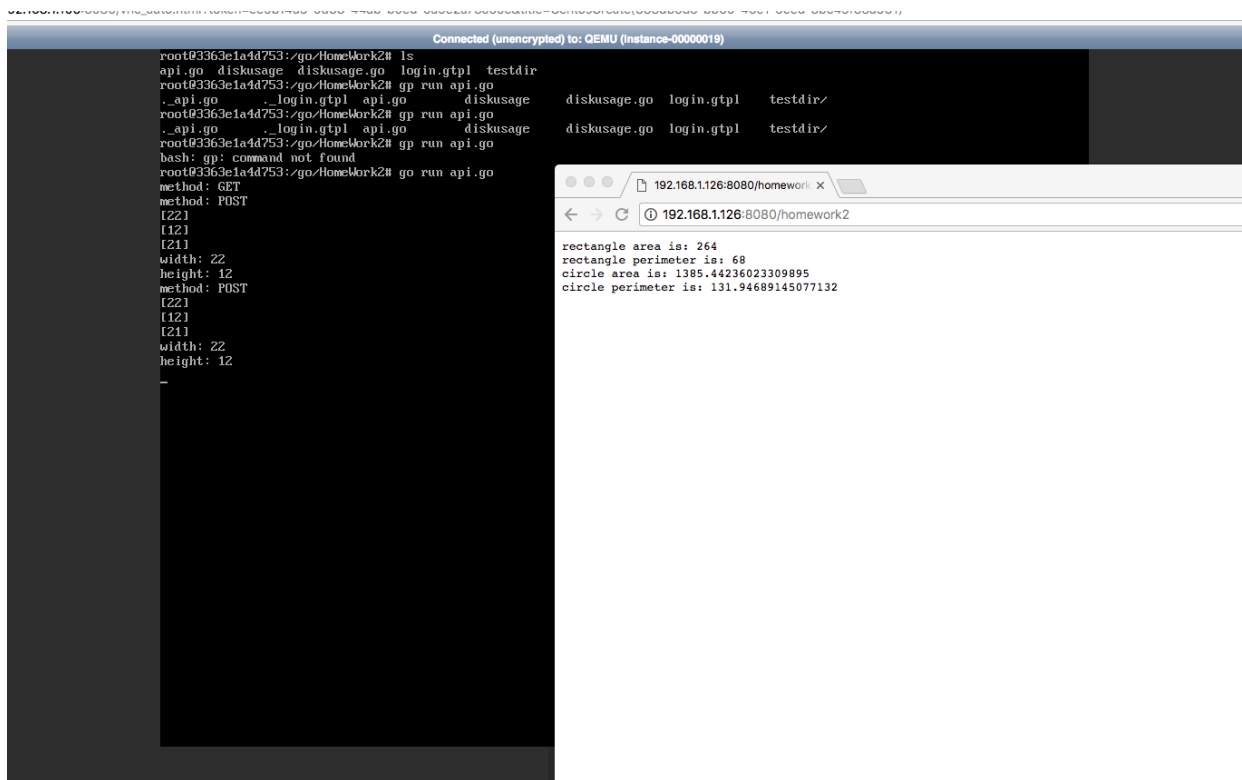
Homwwork2 JieLiu

rectangle

width: 22height: 12

circle

radius: 21result



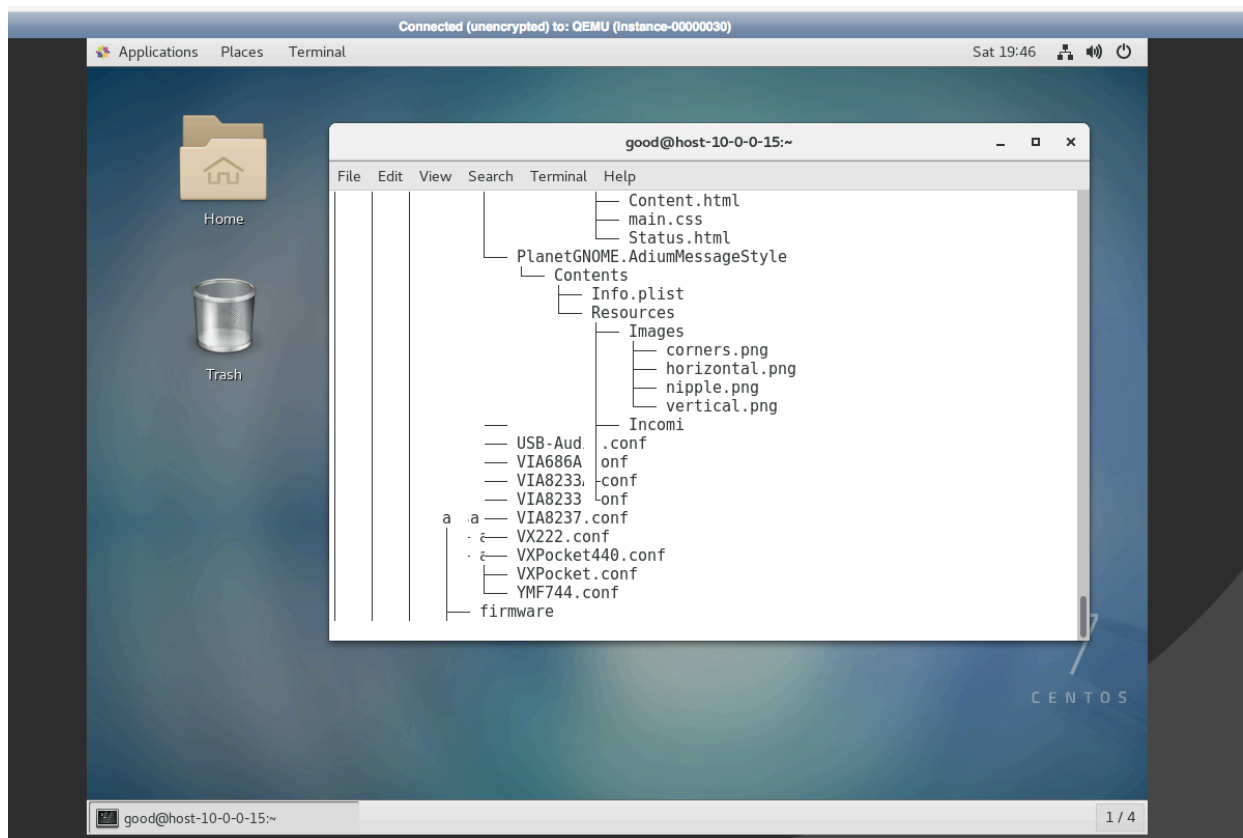
部署一些虚拟机环境测试

windows虚拟机

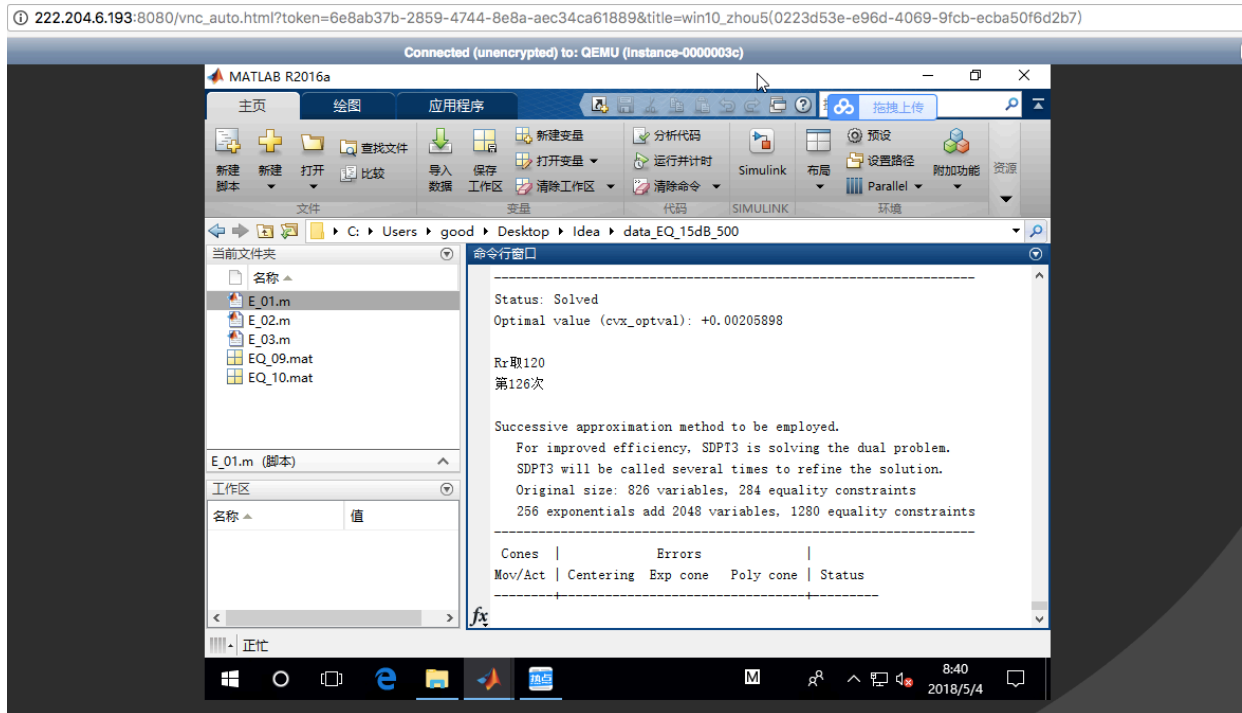


ubuntu虚拟机

centos虚拟机



运行matlab的虚拟机



- 通过脚本创建虚拟机

创建镜像

```
source ./keystonerc_admin
glance image-create --name "Centos7.4" --visibility public --disk-format
qcow2 --container-format bare --file CentOS-7-x86_64-GenericCloud-
1503.qcow2
```

启动虚拟机

```
nova boot --flavor m1.small --image centos-74 --nic net-id=ee145693-d9f5-
46ac-a944-264da0293f31 --availability-zone nova Centos74
```

实验思考

- nova如何调优

`vcpu_pin_set` 建议值是预留前几个物理 CPU，把后面的所有 CPU 分配给虚拟机使用

`cpu_allocation_ratio` = 4.0 物理 CPU 超售比例，默认是 16 倍，超线程也算作一个物理 CPU，需要根据具体负载和物理 CPU 能力进行综合判断后确定具体的配置

`ram_allocation_ratio` = 1.0 内存分配超售比例，默认是 1.5 倍，生产环境不建议开启超售。

`reserved_host_memory_mb` = 4096 内存预留量，这部分内存不能被虚拟机使用

`reserved_host_disk_mb` = 10240 磁盘预留空间，这部分空间不能被虚拟机使用

`service_down_time` = 120 服务下线时间阈值，默认60，如果一个节点上的 nova 服务超过这个时间没有上报心跳到数据库，api 服务会认为该服务已经下线，如果配置过短或过长，都会导致误判。

参考资料

<https://docs.openstack.org/queens>