

# Linux云计算课程笔记

光146班 5号 乔旭 方向：考研

## 目录

## Linux云计算课程笔记

### 目录

#### 一、服务器搭建

- 1.1.前言
- 1.2.配置
- 1.3.服务器基本使用
- 1.2.准备网络和局域网
- 1.3.重定向
- 1.4.启动网站
- 1.5.其它

#### 二、容器

- 2.1. 前言
- 2.2.安装容器
- 2.3.文件系统
- 2.4.开始使用
- 2.5.添加用户
- 2.6.容器基本操作

#### 三、LAMP架构

- 3.1.安装
- 3.2.开始配置
- 3.3.配置Mysql

#### 四、基本Linux命令

- 4.1.基本使用命令
- 4.2.有关打印的相关命令
- 4.3.用户命令详解
- 4.3.shell编程

#### 五、私有云搭建

- 5.1. 说明
  - 十大组件
- 5.2. 部署
  - 1. 创建一个虚拟机
  - 2. 安装和配置
  - 3. 配置网络
  - 4. 配置脚本
  - 5. 安装
  - 6. 运行脚本
  - 7. 运行
  - 8. 补充

#### 六、问题

# 一、服务器搭建

例子：青云（[www.qingcloud.com](http://www.qingcloud.com)）。用则购买，在其中置办主机，网络，硬盘，路由器以及公网IP等。之后在阿里云上进行搭建，包括httpd的基本搭建，以及容器的基本操作。

## 1.1.前言

- 云计算：加速计算。
- 容器技术：无虚拟机。
- 无服务器模式：程序上传代码，然后运行。程序终止，运行结束。（亚马逊已经实现的无服务器模式）
- 边缘计算：每个设备都可以进行自我运算，进行优化。
- 其它：雾计算，霾计算。

### 计算能力

将所有计算机的能力协调使用起来。有弹性地使用计算能力。根据情况进行配置。

### 操作系统

云计算不需要安装操作系统，直接进行硬盘打包，直接使用。安装时间太长。企业常使用的操作系统有CentOS,Ubuntu。

按照时间进行计费，使用结束后直接删掉。即用即买。

云平台的流行，使得服务器的购买市场剧烈缩水。

### 私有云的存在

使用通用的云计算平台，容易造成自己公司的数据流失。

## 1.2.配置

### 1. 设置和购买

- 主机1：One 基本使用（购买）
- 主机2：用于组网-连接Twoswitch（购买）
- 交换机1：Oneswitch
- 交换机2：Twoswitch
- 路由器1：Onerouter（购买）
- 公网IP:（购买）
- 硬盘：Onerisk（购买）

### 2. 网络——基础网络与私有网络

基础网络：所有人公用

私有网络（交换机）：

1. 受管网络：使用路由器来管理私有网络（自动分配IP）
2. 自管网络：自己管理私有网络（自己分配IP）

## 1.3.服务器基本使用

## 1. 使用命令

```
lscpu          --显示配置状态
free -m        --显示内存状态
lsblk          --显示硬盘状态
ping www.baidu.com --进行网络检查
```

(云平台是一种变相管理虚拟机的方法，使用虚拟机只能发挥物理服务器的80%左右的性能，有损失)

## 2. 检查

检查监控

使用自动监控报警系统

会自动对服务器状态进行报警，其中包括网络带宽状态（被攻击），硬盘容量（访问过多）

## 3. 网络

创建交换机Oneswitch（自管）

->创建路由器Onerouter（大型）

->负载均衡器（可选，配置公网阵列）

->公网IP Oneip（四个层次，云平台只需要购买一个；

1. 按照带宽进行计费：流量计费很容易被黑客攻击；

2. BGP多线：防止不同区域运营商的访问延迟；

3. ICP备案：需要22工作日。暂选不需要；

打开路由器界面，图形化，连接之前设定好的网络。

绑定之前购买的公网IP。

# 1.2.准备网络和局域网

```
yum -y install httpd //用来部署网站软件
```

补充：（yellowdog update method）-y 软件安装方法，从而避免深入软件之间的依赖关系  
Apache 由NASA投资；Openstack也是NASA投资

## 1.3.重定向

```
echo "test" > /var/www/index.html
```

1. echo——回音

2. '>' 表明不是标准输出，而是输出重定向

## 1.4.启动网站

```
systemctl start httpd
ss -anput | grep ":80" //打开端口
curl http://localhost // elinks http://localhost
yum -y install elinks
elinks http://localhost//进行端口转发。点击应该修改进行保存
```

## 1.5.其它

### 1. 防火墙

添加防火墙规则，使用http,使用下行规则。端口80，结束80，源IP不填

### 2. 域名

wanwang.aliyun.com 绑定了云上的相应的IP地址  
用CMD 查看

```
nslookup www.kuaike.me
```

### 3. 停止

1. IP地址解绑；
2. 电脑强制关机之后然后删除
3. 私有网络位置要离开路由器，然后再将路由器进行删除；
4. 域名一定要删掉，SEO优化就会出问题；

## 二、容器

硬件虚拟化：基于真实存在的内存，硬盘和网络

软件虚拟化：也叫做“容器技术”

操作系统虚拟化：实际没有操作系统，轻量级虚拟化，工具主要是容器技术

## 2.1. 前言

1. 80端口，443端口的区别，苹果使用443要求使用https；
2. 邓白氏码申请（苹果开发APP需要申请），可以挣钱；
3. 备案：服务器在什么地方就去什么地方去备案。可以看网站最下面。拿小米官网举例子  
京ICP证 京ICP备 京公网安备 京网文（发表一些文章的，新闻稿之类的）  
可通过“站长工具”对网站的备案的可靠性进行检查。同时可以通过备案号看出他的使用时间。
4. EMO服务器（物联网）

## 5. 补充

### 1. 数据库

注意区分数据库和数据库管理软件MySQL

### 2. 容器管理

```
https://my.oschina.net/taogang/blog/778136?utm\_source=tuicool&utm\_medium=referral  
//常用的容器集群平台的比较
```

## 2.2.安装容器

```
yum -y install docker    //安装docker  
systemctl start docker   //启动  
docker version           //查看版本容器
```

容器安装过程如下：

```
[root@i-8xnz1xu0 ~]# systemctl start docker  
[root@i-8xnz1xu0 ~]# docker version  
Client:  
Version:      1.12.6  
API version:  1.24  
Package version: docker-1.12.6-48.git0fdc778.el7.centos.x86_64  
Go version:   go1.8.3  
Git commit:   0fdc778/1.12.6  
Built:        Thu Sep  7 18:00:07 2017  
OS/Arch:      linux/amd64  
  
Server:  
Version:      1.12.6  
API version:  1.24  
Package version: docker-1.12.6-48.git0fdc778.el7.centos.x86_64  
Go version:   go1.8.3  
Git commit:   0fdc778/1.12.6  
Built:        Thu Sep  7 18:00:07 2017  
OS/Arch:      linux/amd64  
[root@i-8xnz1xu0 ~]#
```

```
[root@i-8xnz1xu0 ~]# docker pull docker.io/ansible/centos7-ansible  
Using default tag: latest  
Trying to pull repository docker.io/ansible/centos7-ansible ...  
latest: Pulling from docker.io/ansible/centos7-ansible  
45a2e645736c: Downloading 63.8 MB/70.39 MB  
1c3acf573616: Download complete  
edcb61e55ccc: Download complete  
cbae31bad30a: Download complete  
aacbdb1e2a62: Download complete  
fdeea4fb835c: Downloading 55.12 MB/69.68 MB
```

```
[root@i-8xnz1xu0 ~]# docker images  
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE  
docker.io/ansible/centos7-ansible  latest      688353a31fde     9 months ago    447.2 MB  
[root@i-8xnz1xu0 ~]# docker run it --name=c1 docker.io/ansible/centos7-ansible:latest /bin/bash
```

## 2.3.文件系统

使用 git（英文：饭桶，无用之人），hub（集线器）

```
docker images
docker pull docker.io/ansible/centos7-ansible //下载docker
```

补充：容器的发明是为了更快捷使用

## 2.4.开始使用

```
docker images
docker run -it --name=c1 docker.io/ansible/centos7-ansible:latest /bin/bash
```

容器使用效果如图所示

```
[root@i-8xnz1xu0 ~]# docker ps -all
CONTAINER ID        IMAGE                                COMMAND                  CREATED
STATUS            PORTS              NAMES
aa1bb92baa76       docker.io/ansible/centos7-ansible:latest  "/bin/bash"            35 minutes ago
Up 35 minutes      c1
[root@i-8xnz1xu0 ~]#

[root@i-8xnz1xu0 ~]# docker run -it --name=c1 docker.io/ansible/centos7-ansible:latest /bin/bash
;@aa1bb92baa76:/opt/ansible/ansible[root@aa1bb92baa76 ansible]#
;@aa1bb92baa76:/opt/ansible/ansible[root@aa1bb92baa76 ansible]#
;@aa1bb92baa76:/opt/ansible/ansible[root@aa1bb92baa76 ansible]#
;@aa1bb92baa76:/opt/ansible/ansible[root@aa1bb92baa76 ansible]#
;@aa1bb92baa76:/opt/ansible/ansible[root@aa1bb92baa76 ansible]# cd
;@aa1bb92baa76:~/root@aa1bb92baa76 ~]# ls
anaconda-ks.cfg
;@aa1bb92baa76:~/root@aa1bb92baa76 ~]# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                1
On-line CPU(s) list:   0
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 6
Model name:            QEMU Virtual CPU
Stepping:              3
CPU MHz:               2399.996
BogoMIPS:              4799.99

```

**\*\*注意：必须有镜像，才会有容器。容器就仿佛一个虚拟机**

## 2.5.添加用户

```
useradd tom
grep tom /etc/passwd
docker ps

docker help
docker attach c1
```

## 2.6.容器基本操作

容器关闭: `exit`  
容器打开: 登录之后, 然后`docker ps`查看有几个容器, 然后`docker attach` 编号  
容器删除: `docker rm` (小心删除, 无法撤回)  
容器关闭: `docker stop` (编号)

# 三、LAMP架构

## 3.1.安装

```
yum -y install httpd mysql mariadb mariadb-server php php-mysqlphp (安装有先后顺序)
systemctl start httpd;systemctl enable httpd;systemctl start mariadb
systemctl status httpd
systemctl status mariadb
```

安装成功后如图所示：



```
Verifying : 1:mariadb-libs-5.5.52-1.el7.x86_64
Verifying : 1:mariadb-libs-5.5.52-1.el7.x86_64
```

```
Installed:
  httpd.x86_64 0:2.4.6-67.el7.centos.2
  mariadb-server.x86_64 1:5.5.56-2.el7

  mariadb.x86_64 1:5.5.56-2.el7
  php.x86_64 0:5.4.16-42.el7
```

```
Dependency Installed:
  apr.x86_64 0:1.4.8-3.el7
  httpd-tools.x86_64 0:2.4.6-67.el7.centos.2
  mailcap.noarch 0:2.1.41-2.el7
  perl-Carp.noarch 0:1.26-244.el7
  perl-Compress-Raw-Zlib.x86_64 1:2.061-4.el7
  perl-DBI.x86_64 0:1.627-4.el7
  perl-Encode.x86_64 0:2.51-7.el7
  perl-File-Path.noarch 0:2.09-2.el7
  perl-Filter.x86_64 0:1.49-3.el7
  perl-HTTP-Tiny.noarch 0:0.033-3.el7
  perl-Net-Daemon.noarch 0:0.48-5.el7
  perl-PIRPC.noarch 0:0.2020-14.el7
  perl-Pod-Perldoc.noarch 0:3.20-4.el7
  perl-Pod-Usage.noarch 0:1.63-3.el7
  perl-Socket.x86_64 0:2.010-4.el7
  perl-Text-ParseWords.noarch 0:3.29-4.el7
  perl-Time-Local.noarch 0:1.2300-2.el7
  perl-libs.x86_64 4:5.16.3-292.el7
  perl-parent.noarch 1:0.225-244.el7
  perl-threads.x86_64 0:1.87-4.el7
  php-cli.x86_64 0:5.4.16-42.el7
  apr-util.x86_64 0:1.5.2-6.el7
  libzip.x86_64 0:0.10.1-8.el7
  perl.x86_64 4:5.16.3-292.el7
  perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.el7
  perl-DBD-MySQL.x86_64 0:4.023-5.el7
  perl-Data-Dumper.x86_64 0:2.145-3.el7
  perl-Exporter.noarch 0:5.68-3.el7
  perl-File-Temp.noarch 0:0.23.01-3.el7
  perl-Getopt-Long.noarch 0:2.40-2.el7
  perl-IO-Compress.noarch 0:2.061-2.el7
  perl-PathTools.x86_64 0:3.40-5.el7
  perl-Pod-Escapes.noarch 1:1.04-292.el7
  perl-Pod-Simple.noarch 1:3.28-4.el7
  perl-Scalar-List-Utils.x86_64 0:1.27-248.el7
  perl-Storable.x86_64 0:2.45-3.el7
  perl-Time-HiRes.x86_64 4:1.9725-3.el7
  perl-constant.noarch 0:1.27-2.el7
  perl-macros.x86_64 4:5.16.3-292.el7
  perl-podlators.noarch 0:2.5.1-3.el7
  perl-threads-shared.x86_64 0:1.43-6.el7
  php-common.x86_64 0:5.4.16-42.el7
```

```
Dependency Updated:
  mariadb-libs.x86_64 1:5.5.56-2.el7
```

```
Complete!
```

## 3.2.开始配置

```
cd /var/www/html
ls
vi index.php
//填写
<?php //探测服务器能不能支持php的网站
phpinfo();
?>

curl http://localhost
//查看能不能访问，通过公网IP//
```

再新建一个 index.php

```
<?php
$link=mysql_connect("localhost","root","ABCabc123");
if($link) echo "mariadb is ok"; //如果能出现，说明php能连接数据库
mysql_close();
?>
```

## 3.3.配置Mysql

```
mysql          //连接mysql,即数据库管理工具
mysqladmin -uroot password 'abc123'      //给数据库设置密码
//然后再连接mysql
mysql -uroot -p
//然后输入密码,隐性输入密码
//最后登录数据库
```

然后下载论坛

discuz! 和discourse (未来最好用这个,还可以在手机端用,但是需要用Ruby的环境)

从网上找论坛源码<http://www.discuz.net/forum-10-1.html>

下载 UTF8版本 [http://download.comsenz.com/DiscuzX/3.4/Discuz\\_X3.4\\_SC\\_UTF8.zip](http://download.comsenz.com/DiscuzX/3.4/Discuz_X3.4_SC_UTF8.zip)

直接把文件下载到家目录下

```
cd //到家目录
wget http://download.comsenz.com/DiscuzX/3.4/Discuz_X3.4_SC_UTF8.zip
```

下载后解压出来

```
unzip Discuz_X3.4_SC.UTF8.zip
ls
cd upload
ls
cp -r * /var/www/html
```

安装软件的过程就是复制的过程,此时论坛还没有和mysql结合。

此时打开链接就能看到了,但是没有用户权限。需要以下操作

```
cd /var/www/html
ls
chmod -R 777 config //分配权限
chmod -R 777 data
chmod -R 777 uc_client
chmod -R 777 uc_server
```

然后再打开论坛就可以看见了。其实实际这个过程就是往数据库中写数据的过程。我们下载后其实就已经安装完了,但是数据库中什么都没有而已。

## 四、基本Linux命令

### 4.1.基本使用命令

```
ls      //显示当前目录文件
cd      //切换目录
mkdir   //创建文件夹
touch   //创建文本文件
```

## 4.2.有关打印的相关命令

```
mkdir -p /a/b    //直接创建使用tree /a    可视化的路径结构显示
touch .123       //创建隐藏文件
cat             //全部打印
head            //文件的前几行
tail           //文件的后几行
cat -n passwd | head -7 | tail -1    //只打印文件的第7行
tree /a         //可视化显示文件结构
```

## 4.3.用户命令详解

```
[root@iZwz99zfooa25sc2lifulgZ ~]# ll
总用量 4
-rw-r--r-- 1 root root    0 9月  25 14:36 abc.txt
drwxr-xr-x 2 root root 4096 9月  21 17:21 yantai
```

分为三部分权限，分别是读，写，执行。

```
//设置用户和密码
useradd 用户名
passwd  用户名

//显示用户数目
wc -l /etc/passwd
```

继续查看用户的相关信息

```
[root@iZwz99zfooa25sc2lifulgZ ~]# tail -2 /etc/shadow
dockerroot:!!:17429::::::
qiaoxu:!!:17430:0:99999:7:::
```

可以看出是采用密文加密。其中的17430表示从1970年（计算机元年）至今的天数  
!!表示没有加密

## 4.3.shell编程

shell编程即使用脚本进行编程，用bash进行命令解释，简化计算机管理  
此处创建一个yantai.sh，内容如下：

```
#!/bin/bash      //用bash进行解释给电脑
#print hello world
#tfc@2017.9.25@yd

echo "hello world"

#add useradd
useradd hmm
#
for i in 1..100
do
    useradd eric$i
done
```

可以通过 `ll` 查看文件状态，黑色的表示普通文件，绿色的表示可执行文件  
同时可通过 `chmod` 添加删除用户权限  
代码执行效果如下：

```
执行过程
[root@iZwz99zfooa25sc2lifulgZ ~]# for i in {1..10}
> do
> echo "hello world"
> done
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world

chmod a+x yantai.sh
```

## 五、私有云搭建

兼容服务器的功耗和能力。

IaaS ——基础设置即服务（第一种云计算平台）-> 赚到钱了（取代了硬件服务器提供商）

PaaS ——平台即服务（第二种云计算平台）

Caas ——容器即服务（灵雀云）

FaaS ——无服务器模式

## 5.1. 说明

使用工具：OpenStack

OpenStack：NASA发起的，以Apache许可证授权的自由软件和开放源代码项目。是一个开源的云计算管理平台项目，由几个主要的组件组合起来完成具体工作。支持几乎所有类型的云环境。有很多的企业在支持。使用Python语言编写

## 十大组件

1. 计算：Nova
2. 对象存储：Swift。联想到网盘存储这种模式；
3. 镜像服务：Glance
4. 身份服务：Keystone
5. 网络&地址管理：Neutron
6. 块存储：Cinder
7. UI界面：Horizon
8. 测量：Ceilometer,进行计费 and 监控
9. 部署编排：Heat
10. 数据库服务：Trove

### 版本问题

本次部署使用Newton版本

## 5.2. 部署

### 1. 创建一个虚拟机

在自己的物理机上创建一个虚拟机，使用CentOS7系统。操作如之前所示。

创建两个文件夹，分别存放创建号的虚拟机文件以及OpenStack文件

- **版本**：Newton
- **配置**：
  - 处理器：2个
  - 内存：12G
  - 网络：使用桥接网络（注意勾选连接到物理网络）
  - 磁盘：100G + 20G（专门给别人当网盘用）
  - 网卡：2块（一个自己联网，另一个给别人联网）

——备注：硬件中打开**虚拟化**选项。同时所有操作最好在创建虚拟机时创建完毕，特别是磁盘和网卡。否则之后创建可能不会显示第二块网卡的文件配置信息。

## 2. 安装和配置

1. 联网  
在CentOS创建之初对网络进行配置
2. 安装软件包
3. 通过xshell远程进行登录

## 3. 配置网络

```
ip a s    //查看IP
cd etc/sysconfig/network-config
//打开网卡文件，会发现两块网卡的文件信息。更改第一块网卡的信息。
//配置静态IP
//添加内容，包括IP地址，子网掩码，网关，DNS1，DNS2
//重启网卡
```

## 4. 配置脚本

1. 将已有的OpenStack文件目录中的文件，通过XShell文件传输传送到虚拟机根下。
2. 移动脚本位置到mnt目录中
3. 解压
4. 配置yum仓库，将所有yum源压缩，使用安装包中的yum源。
5. 配置自己的yum源

```
name = repo
baseurl = file :///添加上自己的软件包的路径
enabled = 1    //启动
gpgcheck = 0   //查看文件的md5值，建立gpg的密钥。实现说明，否则后面安装失败
proxy = _none_ //代理
```

### 6. 检查

```
yum repolist    //检查自己的yum是否生效
```

## 5. 安装

1. 进入到OpenStack文件夹，lib文件夹中，配置更改其中的hosts文件和installrc文件
2. 配置 **hosts**  
加入自己的IP地址，以及自己的电脑名字。在该电脑中部署云平台
3. 修改安装环境 **installrc**  
更改密码；修改第二块网卡名（给所有云主机提供网络）；修改硬盘（给所有云主机提供硬

盘)

## 6. 运行脚本

1. `bash` 执行安装，进行OpenStack文件夹（一共8步，依次执行安装）

```
bash main.sh //执行文件
```

2. 依次进行安装，第一步安装后会重启，之后进入后继续第二步安装即可。

## 7. 运行

1. 直接从浏览器上登录，登录IP地址/dashboard/
2. 显示的是openstack的网站界面，显示所有的管理
3. 创建云主机，配置均相同，分别对应着之前设置的硬盘，网卡等
4. 登录使用即可
5. 设置：值得注意的内容包括浮动IP,安全组。
6. 可以使用virsh list 命令显示已经创建的云主机

## 8. 补充

1. python 使用用来快速验证原型，搭建网站，之后用其它语言重写；
2. VCPU：虚拟CPU
3. 浮动IP
4. 镜像可以自己制作，在其中揉入安全组策略等，方便部署；
5. KVM虚拟机进行虚拟化，将自己创建的虚拟机导入

# 六、问题

1. 百万物联网接入系统如何使用？