

第 00 章作业安装与配置 CentOS 7.5

1652817 钟钰琛 计算机科学与技术

2018 年 9 月 21 日

1 本机环境



图 1: 本机环境

2 安装 CentOS 7.5 虚拟机

首先在 VMWare 中建立新的虚拟机, 选择安装 CentOS 7 64 位, 填好相应的配置:

1. 处理器配置: 处理器数量 2, 每个处理器内核数 4
2. 虚拟机内存: 2048MB
3. 网络类型: NAT
4. 磁盘容量: 256G, 因为我的磁盘是 NTFS 文件系统, 所以选择“将虚拟磁盘存储位单个文件”

5. 删除不需要的 USB, 声卡, 打印机设备

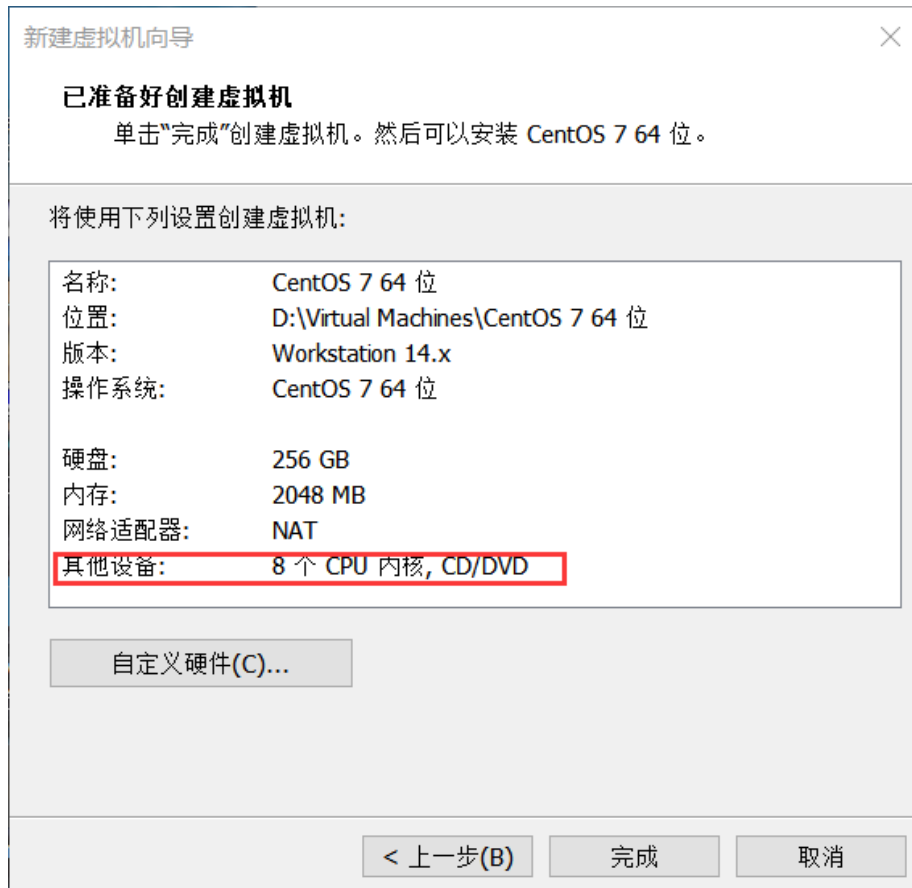


图 2: 创建虚拟机的配置

然后第二步，安装 CentOS 7.5 首先选择中文，自动分区，然后配置网络。把主机名换成 vm-linux 然后因为我的 VMnet8 的 IPv4 设置是“自动获得”所以输入 ipconfig 后，查看到 VMnet8 的 IPV4 地址是 192.168.159.1 所以在 Linux 中，我给了一个 192.168.159.22 这么一个地址。

之后是选择需要安装的软件，按照要求选择了指定的 7 个。最后就是安装，设置 root 密码

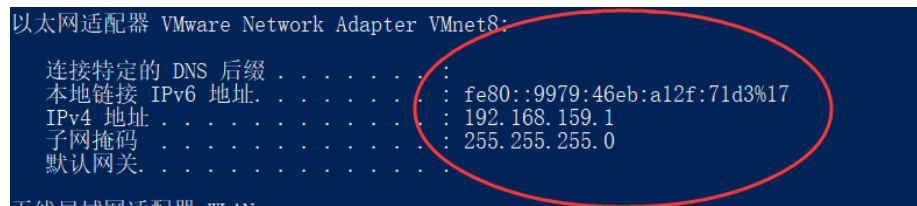


图 3: VMnet8



图 4: Linux 中网络配置



图 5: 安装中

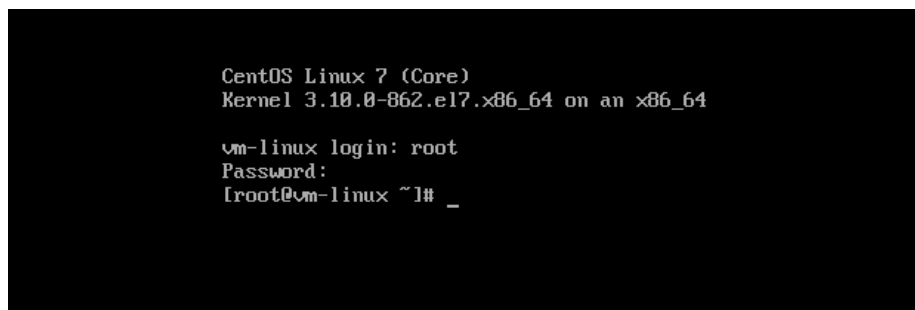


图 6: 成功进入 CentOS 7.5

3 网卡生效

进入后输入 `ifconfig`，发现网卡没有生效。于是首先做些配置，

```
cd /etc/sysconfig/network-scripts
ls -l
vim ifcfg-ens32
```

打开后，首先把 **BOOTPROTO=static**，静态分配。然后设置 **ONBOOT=yes**，自动启动。保存

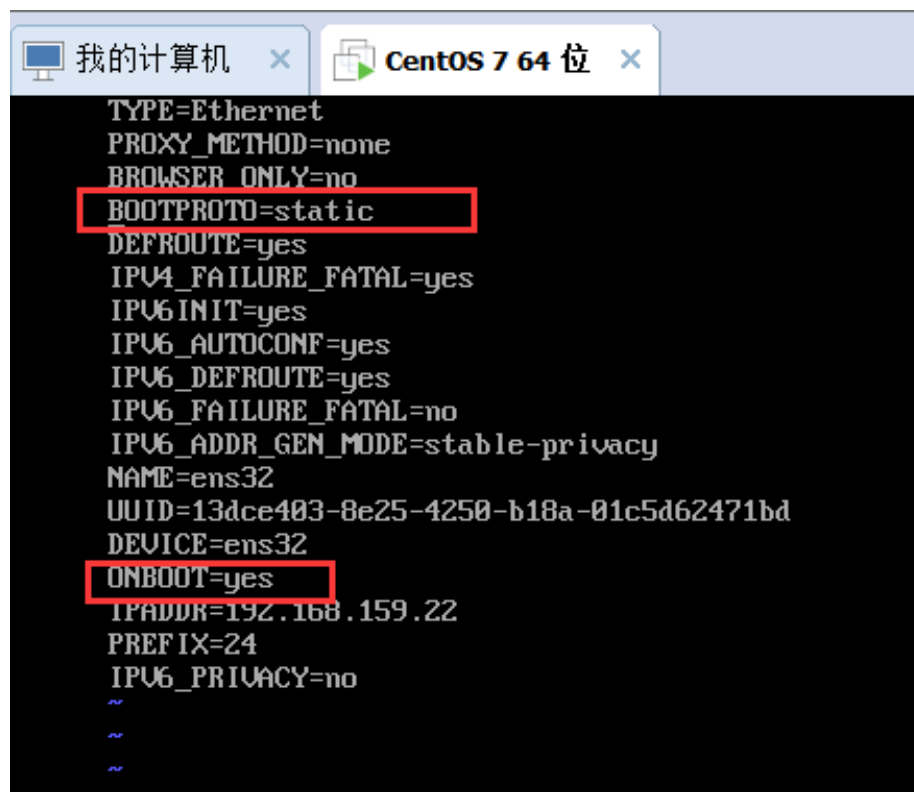


图 7: ifcfg-ens32 配置

然后输入

```
systemctl start network.service
systemctl enable network.service
```

输入 ifconfig 后，终于看到正常结果，且重启后也能自动启动网卡

```
[root@vm-linux network-scripts]# ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.159.22 netmask 255.255.255.0 broadcast 192.168.159.255
    inet6 fe80::f1fc:34ba:b012:0500 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:99:df:a5 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 744 (744.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 200 bytes 16096 (15.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 200 bytes 16096 (15.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@vm-linux network-scripts]#
```

图 8: 网卡生效

4 ping 通

宿主机 (windows) 现在是能直接 ping 通虚拟机 (linux) 的。

```
PS C:\Users\admin> ping 192.168.159.22

正在 Ping 192.168.159.22 具有 32 字节的数据:
来自 192.168.159.22 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.159.22 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.159.22 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.159.22 的回复: 字节=32 时间<1ms TTL=64

192.168.159.22 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 0ms, 平均 = 0ms
PS C:\Users\admin>
```

图 9: 宿主机 ping 通虚拟机

因为已经设置了掩码是 255.255.255.0, 宿主机的 IPv4 地址是 192.168.159.1,

虚拟机是 192.168.159.22，所以已经在同一个网段
虚拟机尚未能 ping 通宿主机

5 SecureCRT 登陆

能直接登陆

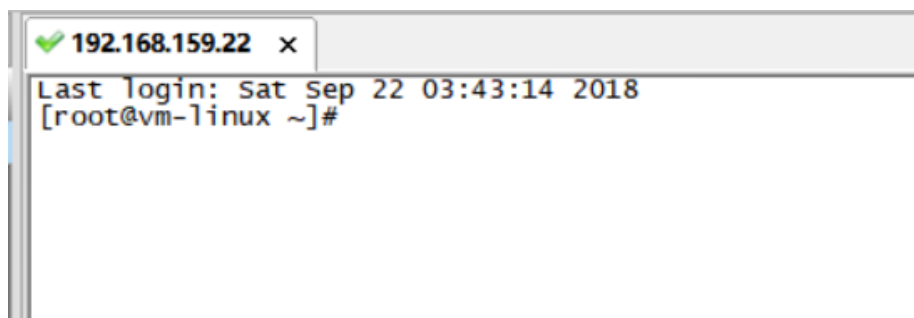


图 10: SecureCRT 登陆

6 显示中文

修改/etc/locale.conf 文件，修改 LANG="zh_CN.gbk"，保存



图 11: locale.conf

重启后中文正常

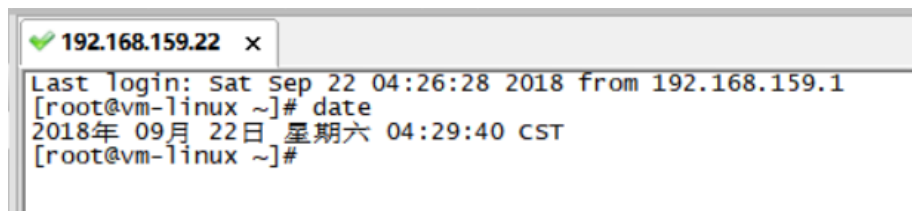


图 12: 中文正常

7 数据库

首先启动 MariaDB, 设置开机启动, 然后进行相关配置

```
systemctl start mariadb
systemctl enable mariadb
mysql_secure_installation
```

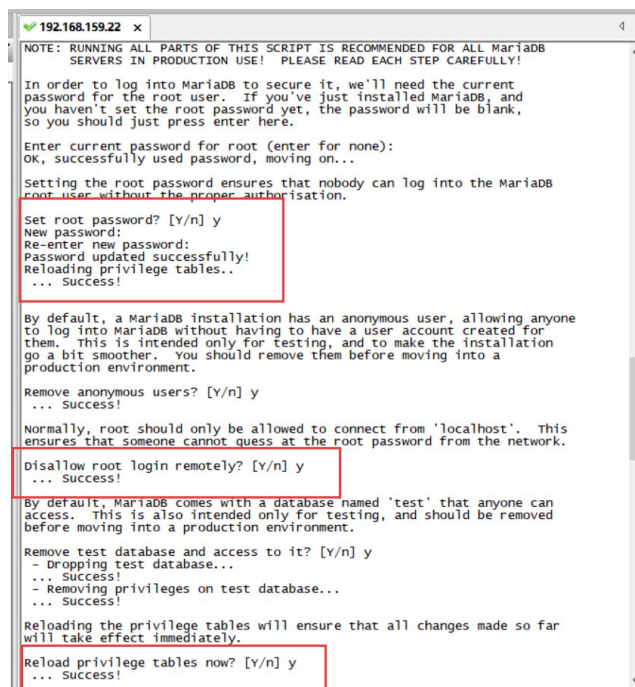
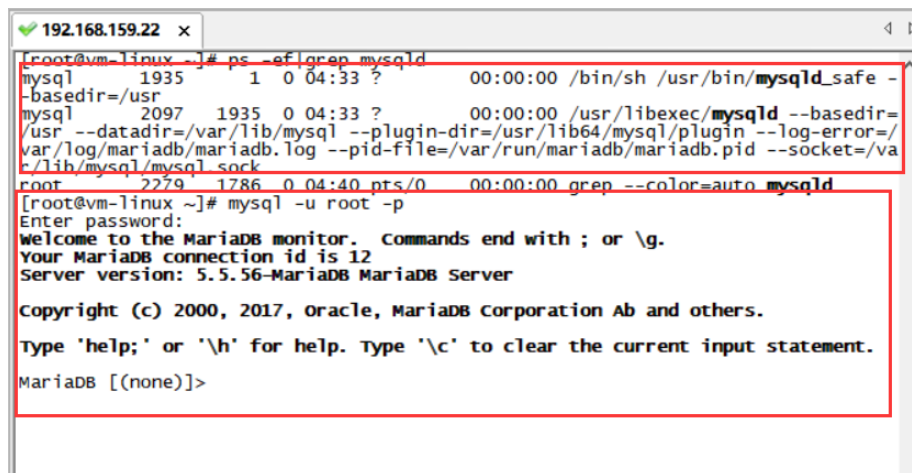


图 13: 安装配置

A terminal window titled '192.168.159.22 x' showing the process of starting MySQL. The user runs 'ps -ef | grep mysqld' and sees two processes: one for 'mysqld_safe' and one for 'mysqld'. Then, the user runs 'mysql -u root -p', enters a password, and is greeted by the MySQL monitor. The terminal output is as follows:

```
[root@vm-linux ~]# ps -ef | grep mysqld
mysql      1935      1  0 04:33 ?        00:00:00 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
mysql      2097     1935  0 04:33 ?        00:00:00 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib64/mysql/plugin --log-error=/var/log/mariadb/mariadb.log --pid-file=/var/run/mariadb/mariadb.pid --socket=/var/lib/mysql/mysql.sock
root       2279     1786  0 04:40 pts/0    00:00:00 grep --color=auto mysqld

[root@vm-linux ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

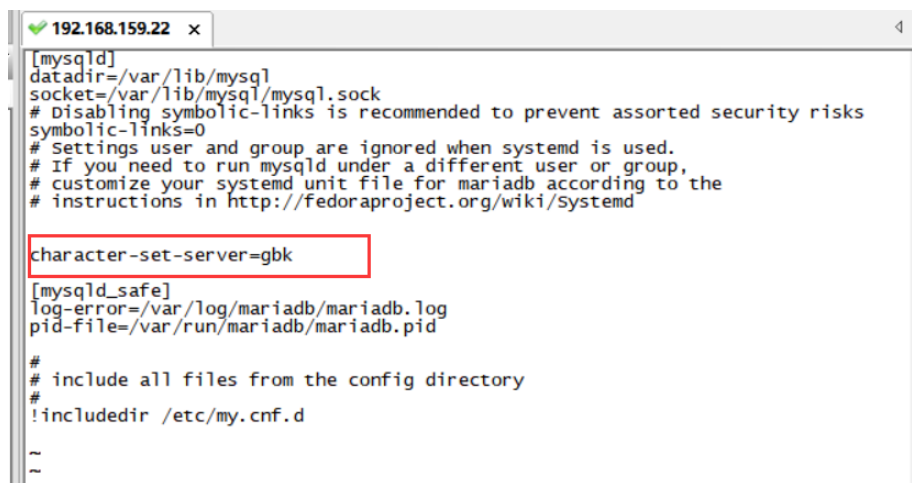
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

图 14: 数据库启动与进入成功

8 MySQL 支持中文

先关闭 mariadb,然后在/etc/my.cnf 里面添加一句 character-set-server=gbk

A terminal window titled '192.168.159.22 x' showing the contents of the MySQL configuration file /etc/my.cnf. The file contains various settings for the MySQL server, including datadir, socket, symbolic-links, and character-set-server. The line 'character-set-server=gbk' is highlighted with a red box. The terminal output is as follows:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

character-set-server=gbk

[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid

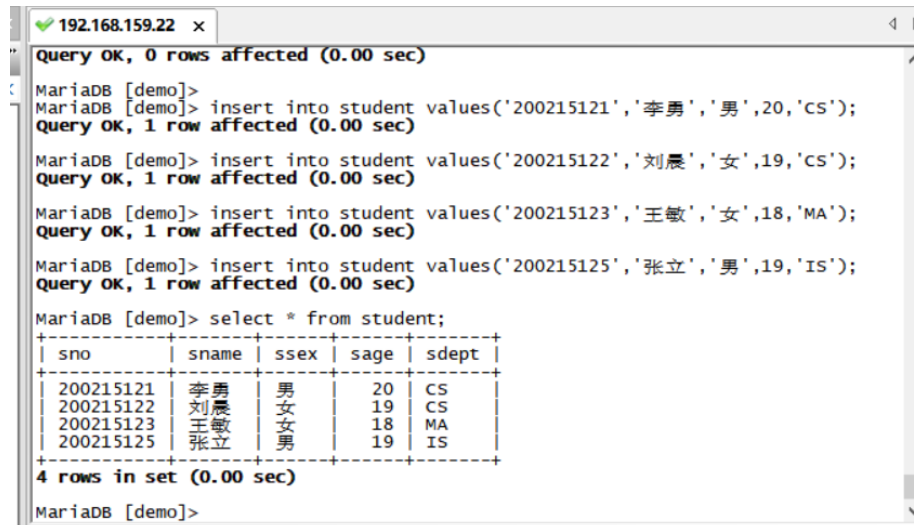
#
# include all files from the config directory
#
!includedir /etc/my.cnf.d

~
~
```

图 15: /etc/my.cnf

重启开启 mariadb, 输入 sql

可以看到中文正常



```
192.168.159.22 x
Query OK, 0 rows affected (0.00 sec)

MariaDB [demo]>
MariaDB [demo]> insert into student values('200215121','李勇','男',20,'CS');
Query OK, 1 row affected (0.00 sec)

MariaDB [demo]> insert into student values('200215122','刘晨','女',19,'CS');
Query OK, 1 row affected (0.00 sec)

MariaDB [demo]> insert into student values('200215123','王敏','女',18,'MA');
Query OK, 1 row affected (0.00 sec)

MariaDB [demo]> insert into student values('200215125','张立','男',19,'IS');
Query OK, 1 row affected (0.00 sec)

MariaDB [demo]> select * from student;
+-----+-----+-----+-----+-----+
| sno   | sname | ssex | sage | sdept |
+-----+-----+-----+-----+-----+
| 200215121 | 李勇  | 男   | 20   | CS    |
| 200215122 | 刘晨  | 女   | 19   | CS    |
| 200215123 | 王敏  | 女   | 18   | MA    |
| 200215125 | 张立  | 男   | 19   | IS    |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MariaDB [demo]>
```

图 16: mysql 中文正常

9 PHP 服务

到这一步，需要用 yum 来安装各种库了，因为不能上外网，所以需要挂载 iso

首先建立两个文件夹

```
mkdir /mnt/iso
mkdir /mnt/cdrom
```

然后用 ftp 传 iso 文件放到/mnt/cdrom 里面

然后挂载 iso 镜像到/mnt/cdrom 下

```
mount -o loop /mnt/iso/CentOS-7-x86_64-DVD-1804.iso /mnt/cdrom
```

用 df -h 来查看是否挂载成功

然后改名/etc/yum.repos.d/目录下所有文件 (我是在后面加.old)，然后建立一个“Centos-ISO.repo”文件里面写入：

最后执行

```
[root@vm-linux ~]# df -h
文件系统 容量 已用 可用 已用% 挂载点
/dev/mapper/centos-root 50G 6.1G 44G 13% /
devtmpfs 899M 0 899M 0% /dev
tmpfs 911M 0 911M 0% /dev/shm
tmpfs 911M 9.5M 902M 2% /run
tmpfs 911M 0 911M 0% /sys/fs/cgroup
/dev/sda1 1014M 145M 870M 15% /boot
/dev/mapper/centos-home 203G 33M 203G 1% /home
tmpfs 183M 0 183M 0% /run/user/0
/dev/loop0 4.2G 4.2G 0 100% /mnt/cdrom
[root@vm-linux ~]#
```

图 17: 挂载成功



```
[base]
name=CentOS-$releasever - Base
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&rep
o=os&infra=$infra
baseurl=file:///mnt/cdrom
enabled=1
gpgcheck=0
gpgkey=file:///mnt/cdrom/RPM-GPG-KEY-CentOS-7
~
```

图 18: CentOS-ISO.repo

```
yum clean all
yum update all
```

就可以使用了

首先需要安装 Apache

```
yum -y install httpd
```

然后开启 apache 服务, 并开机自启

```
systemctl start httpd.service
systemctl enable httpd.service
```

并且设置防火墙 (不然会出现 “无法访问”)

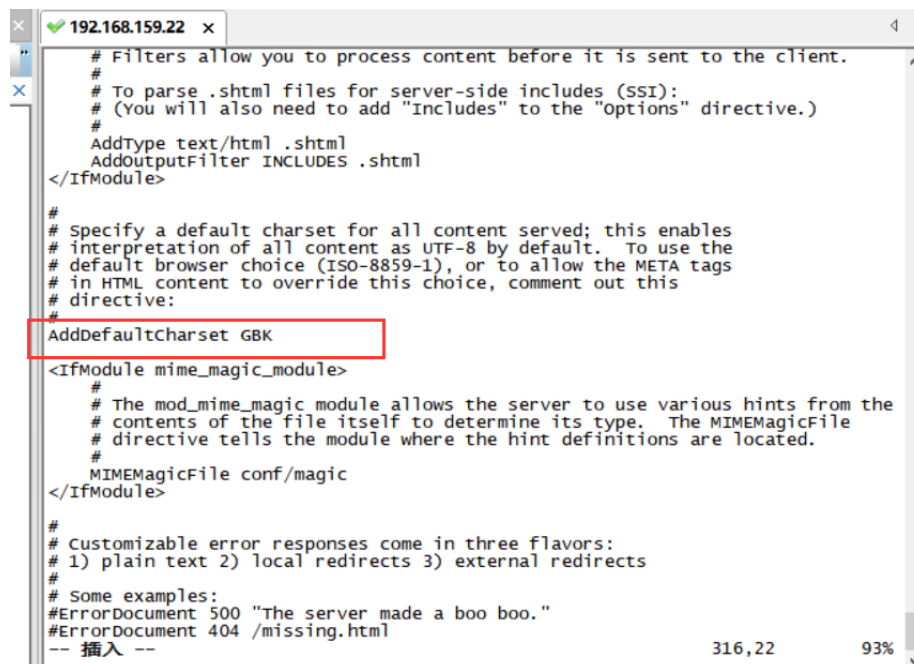
```
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --reload
```

第二步, 为了使用 php, 并且与 mysql 交互, 需要安装这些东西

```
yum -y install php php-mysql
```

然后重启 httpd 服务

第三步, 需要修改 Apache 设置, 防止乱码. 打开 /etc/httpd/conf/httpd.conf
输入 /AddDefaultCharset 搜索该字段, 修改为 GBK



```
# Filters allow you to process content before it is sent to the client.
#
# To parse .shtml files for server-side includes (SSI):
# (You will also need to add "Includes" to the "Options" directive.)
#
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
</IfModule>

#
# Specify a default charset for all content served; this enables
# interpretation of all content as UTF-8 by default. To use the
# default browser choice (ISO-8859-1), or to allow the META tags
# in HTML content to override this choice, comment out this
# directive:
#
AddDefaultCharset GBK

<IfModule mime_magic_module>
#
# The mod_mime_magic module allows the server to use various hints from the
# contents of the file itself to determine its type. The MIMEMagicFile
# directive tells the module where the hint definitions are located.
#
MIMEMagicFile conf/magic
</IfModule>

#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
-- 插入 --
```

图 19: httpd.conf

保存, 重启服务

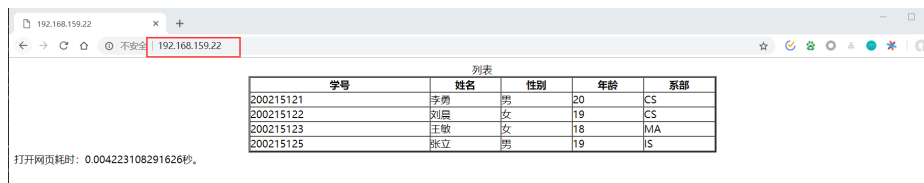
这时候在 /var/www/html/ 目录下建立 demo.php 把里面密码改成自己的密码 “123456”

为了不显示 “demo.php”, 只需要设置成 index 就可以了 保存, 重启服务

打开浏览器输入 “http://192.168.159.22”, 显示正常!

```
#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.html demo.php
</IfModule>
```

图 20: httpd.conf



学号	姓名	性别	年龄	系部
200215121	李勇	男	20	CS
200215122	刘晨	女	19	CS
200215123	王敏	女	18	MA
200215125	张立	男	19	IS

打开网页耗时: 0.004223108291626秒。

图 21: demo.php

10 编译 C++ mysql 程序

首先需要安装

```
yum -y install mysql-devel
```

然后找到 mysql.h 的位置

```
[root@vm-linux ~]# find /usr -name "mysql.h"
/usr/include/mysql/mysql.h
```

建立 demo.cpp 文件，把里面 mysql 密码修改成 “123456”

然后输入如下命令:

```
g++ demo.cpp -o demo -I /usr/include/mysql/ -L
/usr/lib64/mysql/ -lmysqlclient
```

直接编译成功!

```
[root@vm-linux ~]# ls
anaconda-ks.cfg demo demo.cpp
[root@vm-linux ~]# ./demo
select return 4 records
学号: 200215121 姓名: 李勇 性别: 男 年龄: 20 系部: CS
学号: 200215122 姓名: 刘晨 性别: 女 年龄: 19 系部: CS
学号: 200215123 姓名: 王敏 性别: 女 年龄: 18 系部: MA
学号: 200215125 姓名: 张立 性别: 男 年龄: 19 系部: IS
[root@vm-linux ~]#
```

图 22: 运行 demo

11 FlashFXP 传输与中文显示

进入 FlashFXP, 建立连接后, 点击工具栏的“站点”->“站点管理器”->“选项”

修改成如图所示:



图 23: 修改 FlashFXP 配置

然后传几个文件验证下:

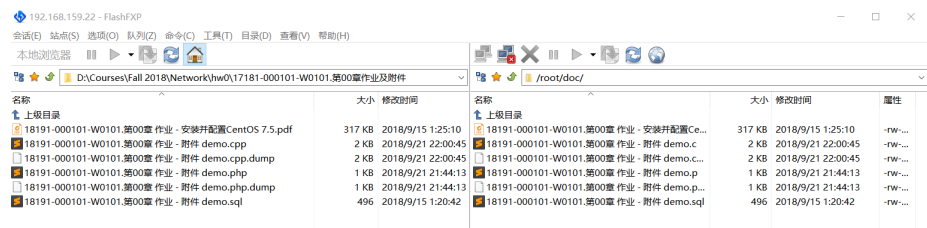


图 24: FlashFXP 显示正常

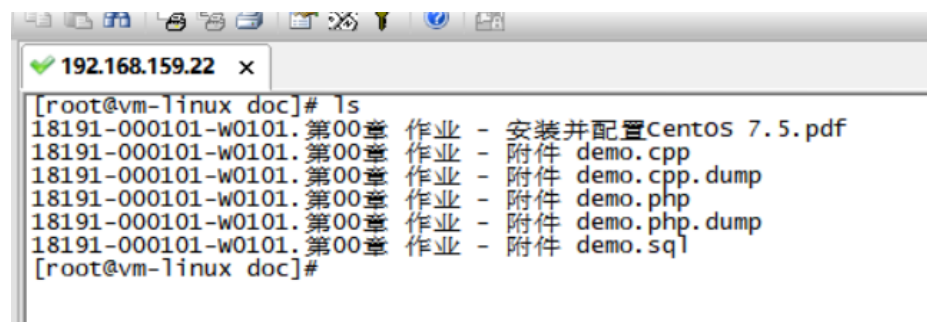


图 25: SecureCRT 显示正常