Linux(RHEL 7/CentOS 7)实训帮助手册

0x01.Linux 远程登陆配置

①开启 sshd 服务

★ # service sshd start　　//开启 ssh 服务

★ # service sshd status　　//查看 ssh 服务状态

```
[root@localhost ~]# service sshd start
Redirecting to /bin/systemctl start  sshd.service
```

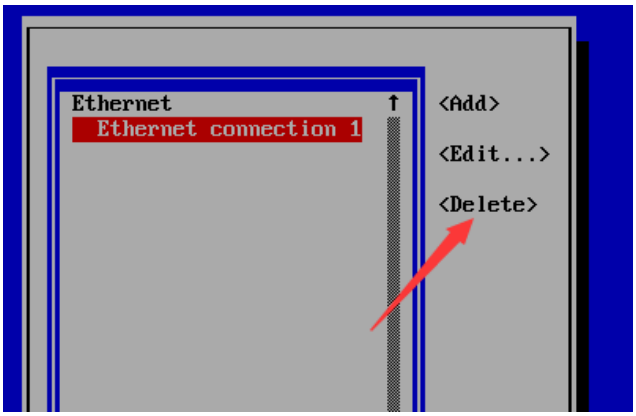
```
[root@localhost ~]# service sshd status
Redirecting to /bin/systemctl status  sshd.service
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; disabled)
   Active: active (running) since Mon 2018-12-10 02:55:26 PST; 6min ago
  Process: 2454 ExecStartPre=/usr/sbin/sshd-keygen (code=exited, status=0/SUCCE
S)
 Main PID: 2455 (sshd)
   CGroup: /system.slice/sshd.service
           └─2455 /usr/sbin/sshd -D

Dec 10 02:55:26 localhost.localdomain systemd[1]: Started OpenSSH server daemon
Dec 10 02:55:27 localhost.localdomain sshd[2455]: Server listening on 0.0.0.0..
Dec 10 02:55:27 localhost.localdomain sshd[2455]: Server listening on :: port..
Dec 10 03:00:25 localhost.localdomain sshd[2550]: Accepted password for root ..
Hint: Some lines were ellipsized, use -l to show in full.
```

②设置 IP 地址

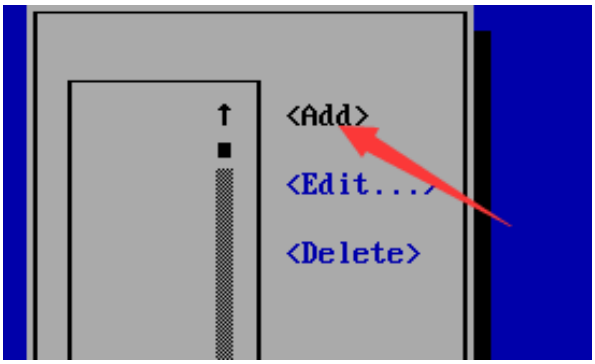
★ # nmtui　　//进入可视化页面设置 ip

1.

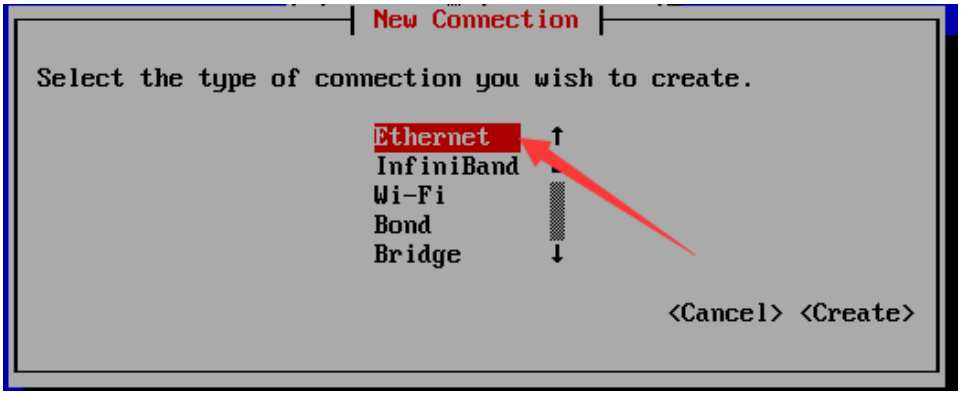
2.删除原有网络设置


Ethernet
Ethernet connection 1
<Add>
<Edit...>
<Delete>

3.


<Add>
<Edit...>
<Delete>

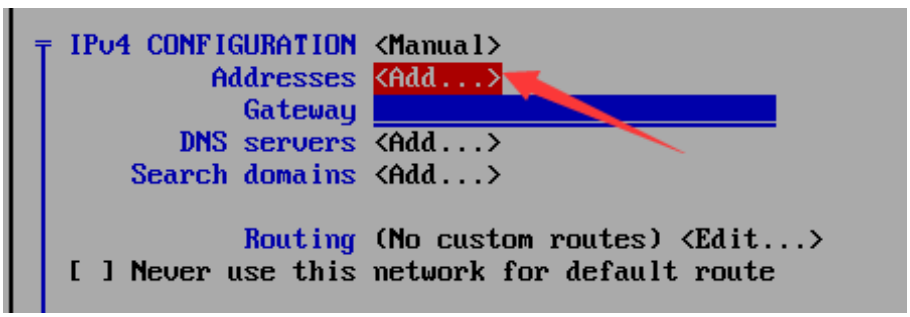
4.


New Connection
Select the type of connection you wish to create.
Ethernet
InfiniBand
Wi-Fi
Bond
Bridge
<Cancel> <Create>

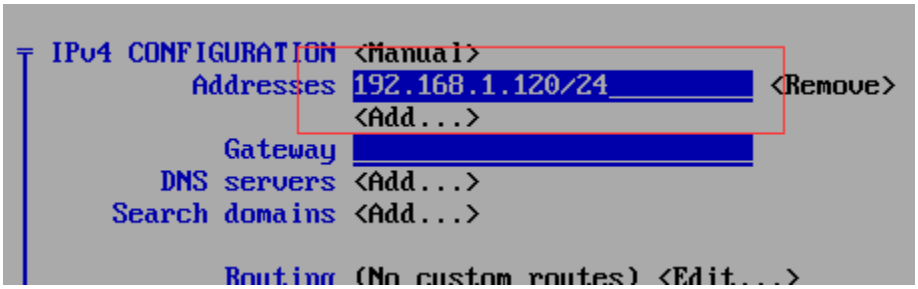
5.


Disabled
Automatic
Link-Local
Manual
Shared
2
Profile name Ethernet connection 1
Device
= ETHERNET
= IPv4 CONFIGURATION <Automatic>
= IPv6 CONFIGURATION <Automatic>
[X] Automatically connect
[X] Available to all users
1

6.

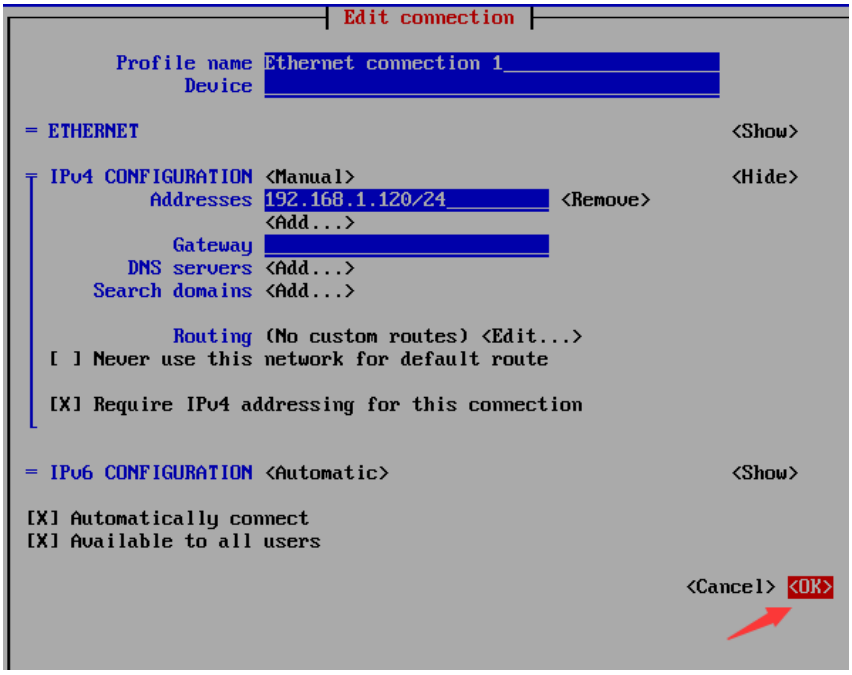

= IPv4 CONFIGURATION <Manual>
Addresses <Add...>
Gateway
DNS servers <Add...>
Search domains <Add...>
Routing (No custom routes) <Edit...>
[ ] Never use this network for default route

7.根据自己 ip 情况 输入 ip 及掩码



(上图为根据笔者自己家里的网段写的,不要直接抄)

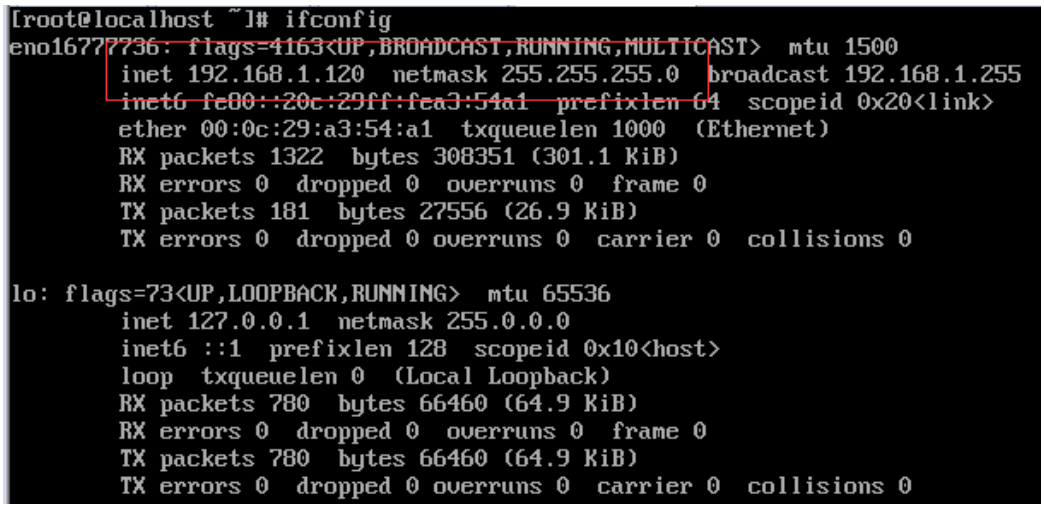
8.



退出 <Quit>

★ # ifconfig //验证设置的 ip 地址是否生效

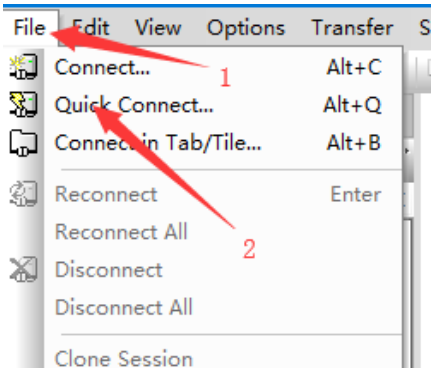
下图为设置成功



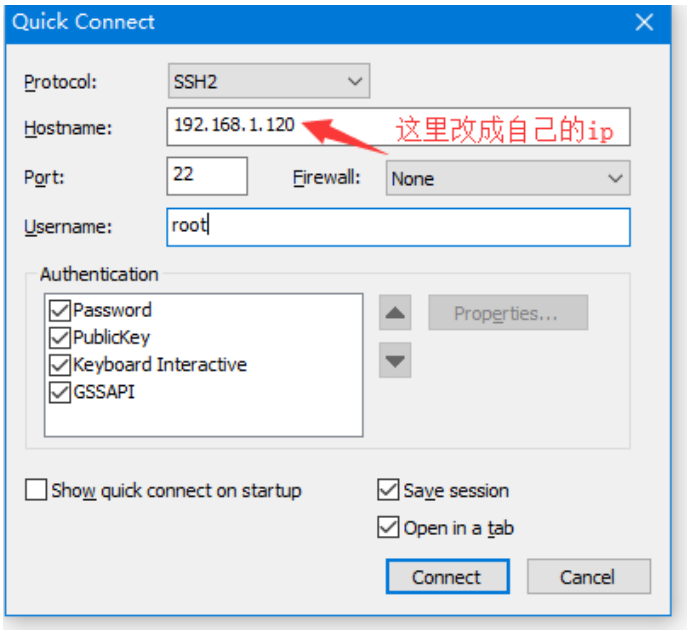
③使用 SecureCRT 软件远程登陆

1. 安装并打开 SecureCRT 软件 

2.

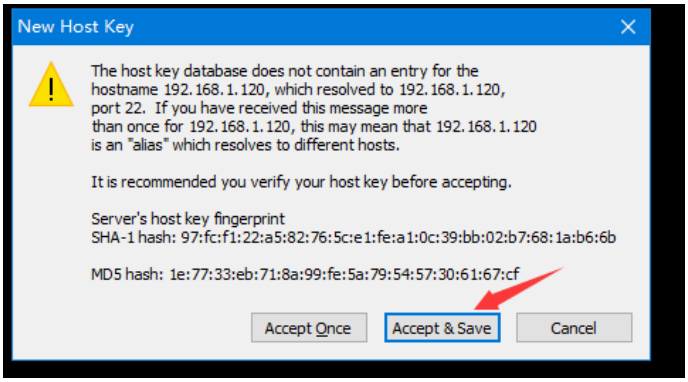


3.

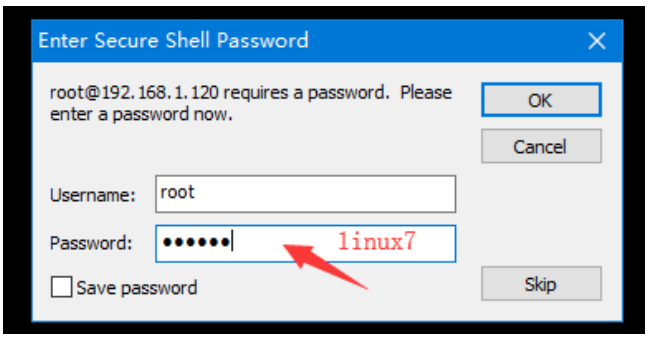


connect

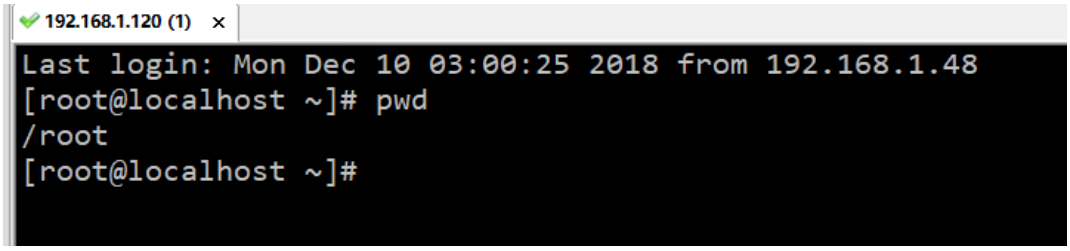
4.



5.



6.



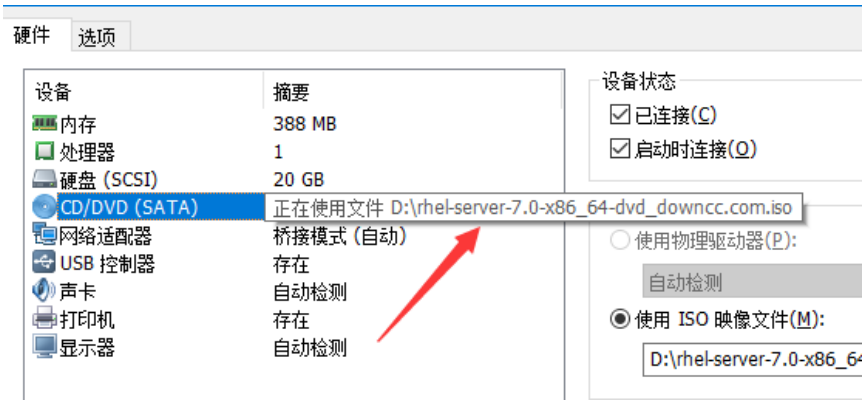
连接成功^_^

0x02.挂载 yum 源

提供两个选择

①挂载光驱(虚拟)

设置虚拟机

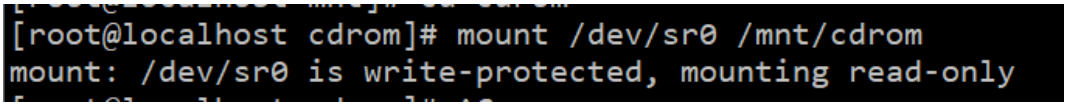


##注意右上已连接

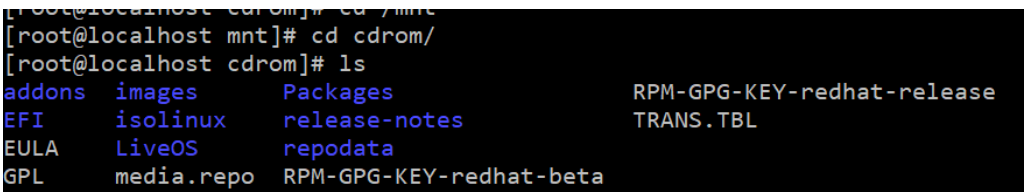
★ # cd /mnt

mkdir cdrom　　//建立挂载到的目录

mount /dev/sr0 /mnt/cdrom　　//挂载



★ # ls cdrom/　　//下图为挂载成功



修改 yum 源

★ # cd /etc/yum.repos.d/　　//记不住可以使用 TAB 键进行补全

vim yum.repo

修改为下图内容

```
[InstallMedia]
name=rhel7.cdrom
baseurl=file:///mnt/cdrom/
enable=1
gpgcheck=0
```

ESC　:wq　　//保存退出

②FTP yum 源

只需将上图 baseurl 的值改为

★ baseurl= ftp://192.168.9.201/ftpfile/RHEL7/

0x03.*了解 rpm 安装　　//此部分了解！可根据实际情况掌握

cd /mnt/cdrom/Packages　　//进入文件目录

ls | grep +包名　　//查找需要安装的包

#rpm -ivh +包名　　//安装　(但是又有联合安装的问题容易尬住所以这里了解)

0x04. 安装 iptables 防火墙

★ # yum install iptables*　　//安装防火墙

service iptables start　　//开启防火墙

service iptables status　　//查看防火墙工作状态

```
[root@localhost ~]# service iptables status
Redirecting to /bin/systemctl status  iptables.service
iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service;
   Active: inactive (dead) since Sun 2018-12-16 16:35:42 PST;
 Main PID: 5071 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/iptables.service
```

0x05.配置 Apache 服务

★ # yum install httpd*

```
Installed size: 16 M
Is this ok [y/d/N]: y
```

中途输入 y 回车

```
Complete!
```

提示安装成功

★ # service httpd start　　//开启 apache

service httpd status　　//查看是否开启成功

```
ttpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system
   Active: active (running) since Mon 2018
Main PID: 30168 (httpd)
```

开启成功

在学习阶段所以为了方便可以关闭防火墙

★ # service iptables start

service iptables stop　　//关闭防火墙

service iptables status　　//查看状态

```
   Loaded: loaded (/usr/lib/sys
   Active: inactive (dead)
```

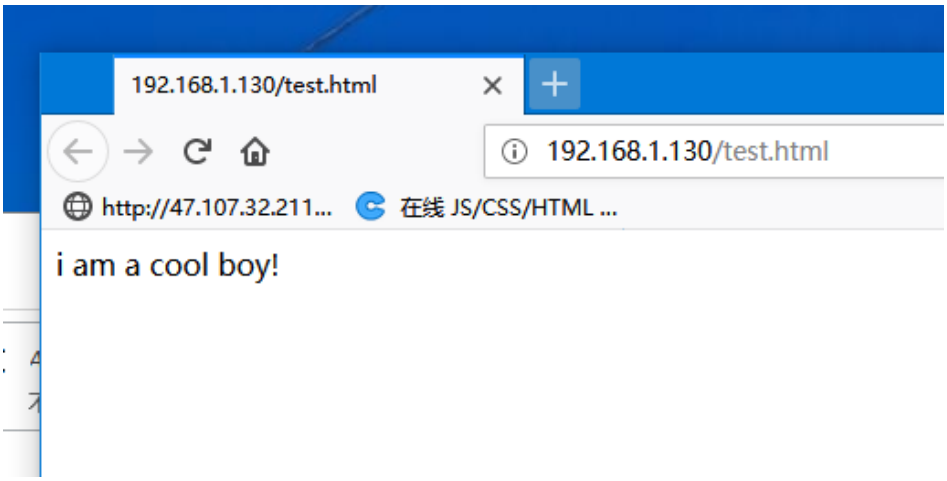
测试 Apache

★ # cd /var/www/html　　//进入网站根目录

vi test.html

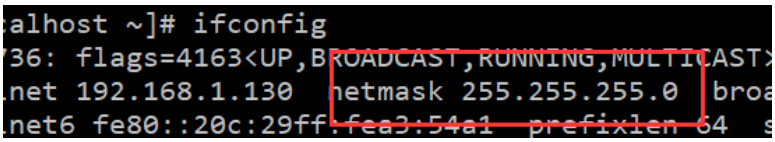
随意输入一些内容　　ESC　　:wq　　保存退出

进入物理机(windows 系统)中的浏览器输入 URL 查看



注:如遇到无法访问的问题,请检查防火墙是否关闭

或者输入 #ifconfig 查看 ip 设置是否有误



红框位置也要注意是否与我相同

0x06. 搭建 Samba 服务器(实现 SMB 协议的免费软件)

★ # yum install samba* //安装 samba

service smb start //开启

service smb status //查看是否成功

```
[root@localhost ~]# service  smb start
Redirecting to /bin/systemctl start  smb.service
[root@localhost ~]# service  smb status
Redirecting to /bin/systemctl status  smb.service
smb.service - Samba SMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/smb.service; disabled)
   Active: active (running) since Tue 2018-12-11 02:33:04 PST; 2s ago
 Main PID: 3008 (smbd)
   Status: "smbd: ready to serve connections..."
   CGroup: /system.slice/smb.service
           ├─3008 /usr/sbin/smbd
           └─3009 /usr/sbin/smbd
```

接下来修改配置文件

★ # vim /etc/samba/smb.conf

进去别动

按 /secur 回车 //这里进行检索

```
        # log files split per-machine:
        log file = /var/log/samba/log.%m
        # maximum size of 50KB per log file, then rotate:
        max log size = 50

# ----------------------- Standalone Server Options ------------------
#
# security = the mode Samba runs in. This can be set to user, share
# (deprecated), or server (deprecated).
#
# passdb backend = the backend used to store user information in. New
# installations should use either tdbsam or ldapsam. No additional con
# is required for tdbsam. The "smbpasswd" utility is available for bac
# compatibility.
#
        security = user
        passdb backend = tdbsam
                                                              114,3
```

找到红框

按 i 键

进入编辑模式

```
    security = user
    passdb backend = tdbsam
    map to guest = bad user
```

改成上图这样 √

按 ESC

按 Shift + g //目标文件尾

按 i 键 并在文件尾输入

```
;       write list = +staff
[share]
        comment = tmp share
        path = /share
        writeable =yes
        browseable =yes
        guest ok =yes
-- INSERT --
```

ESC :wq 保存退出

这里结束后可以# testparm 测试 smb.conf 是否输入正确

★ # mkdir /share/ //建立路径

#chmod -R 777 /share/ //赋予开放权限

关闭 SELinux

★ #setenforce 0 //临时关闭

#sestatus //查看状态

可以修改/etc/sysconfig/selinux 文件进行永久禁用

SELINUX=disabled

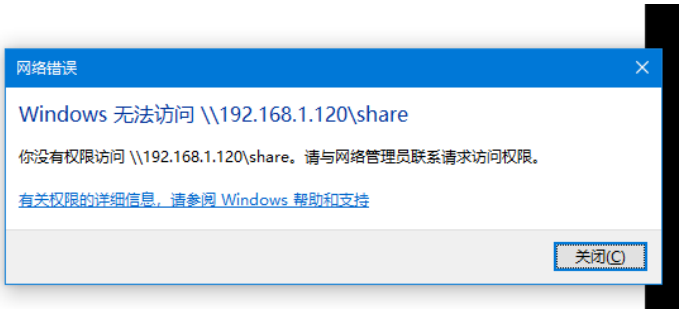
★ # service smb restart //重启 samba 服务器确保修改配置生效

物理机上打开我的电脑

地址栏输入 \\192.168.1.120 //对应你自己的 linux ip 地址

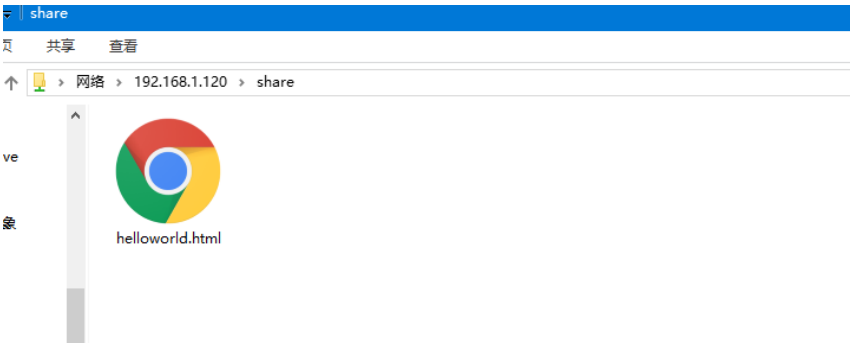
上图为成功配置

如出现下图



重新尝试关闭 SELinux 和防火墙的步骤即可解决

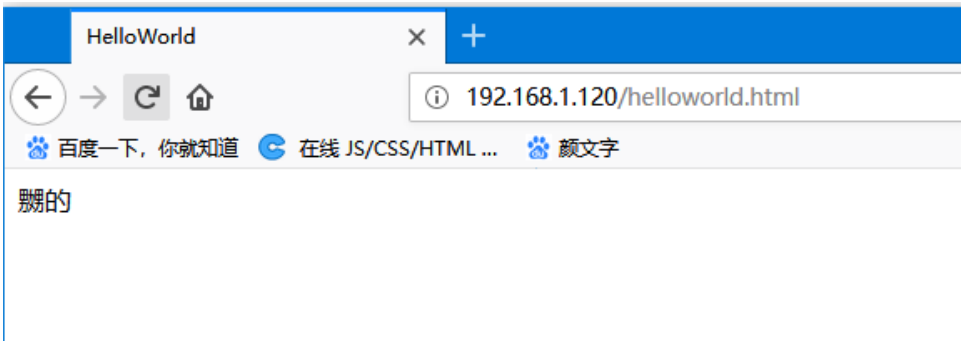
测试： 上传一个网页文件



★ # cp /share/helloworld.html /var/www/html/ 复制过去

或者直接

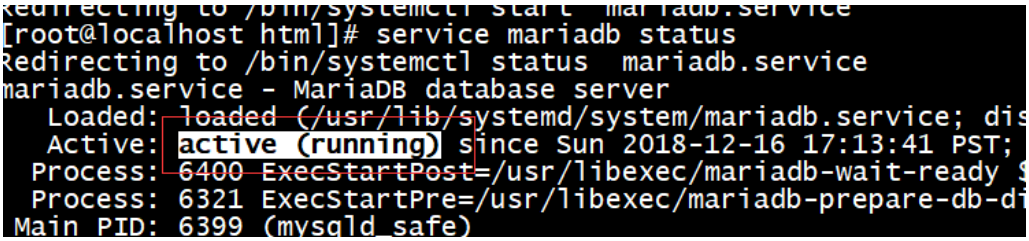
\cp -rf /share/. /var/www/html 完全覆盖掉



0x07.mysql 基础

安装并启动 mariadb

```
# yum install mariadb*          //安装
# service mariadb start        //启动数据库
# service mariadb status       //查看状态
```



这里可以输入 # mysql 进入数据库

①对数据库的操作

show databases;　　//查看现有的库

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.01 sec)

MariaDB [(none)]>
```

create database 数据库名;　　//创建数据库

```
ariaDB [(none)]> create database jsj;
uery OK, 1 row affected (0.01 sec)

ariaDB [(none)]>
```

use 数据库名;　　//选择要使用的库

```
MariaDB [(none)]> use jsj;
Database changed
MariaDB [jsj]>
```

drop database 数据库名;　　//删除库

drop table 表名;　　//删除表

②对数据表的操作

create table 表名　　//创建表

(

列名 1 类型,

列名 2 类型

);

```
MariaDB [jsj]> create table stu
    -> (
    -> name varchar(15),
    -> age int
    -> )default charset=utf8;
Query OK, 0 rows affected (0.02 sec)

MariaDB [jsj]>
```

建表同时设置了编码

可使用 primary key column_name 设置主键

单个列作为主键

```
Database changed
MariaDB [stumanage]> create table stuinfo
    -> (
    -> number char(4) primary key,
    -> age int,
    -> name varchar(15),
    -> sex varchar(8),
    -> addr varchar(20)
    -> );
```

联合主键

```
MariaDB [stumanage]> create table grade
    -> (
    -> score int,
    -> kw char(2),
    -> number char(4),
    -> primary key(kw,number)
    -> );
```

show tables;　　//查看所选库中的表

```
MariaDB [jsj]> show tables;
+---------------+
| Tables_in_jsj |
+---------------+
| stu           |
+---------------+
1 row in set (0.00 sec)
```

插入数据

insert into tablename(列名 1,列名 2,…) values(’值 1’,’值 2’);

insert into tablename values(’值 1’,’值 2’);　//按定义顺序进行插入

```
MariaDB [jsj]> insert into stu values('拉克丝',22);
Query OK, 1 row affected (0.01 sec)
```

更新表中数据

update 表名

set column_name= new_value

where column_name= some_value;

```
MariaDB [jsj]> update stu
    -> set age=23
    -> where name='拉克丝';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [jsj]>
```

查询

select column_name(s) from table_name where 加条件

```
MariaDB [jsj]> select * from stu;
+--------------+------+
| name         | age  |
+--------------+------+
| 拉克丝       |   23 |
| 伊泽瑞尔     |   24 |
+--------------+------+
2 rows in set (0.00 sec)
```

//查询所有

```
MariaDB [jsj]> select * from stu where age=24;
+--------------+------+
| name         | age  |
+--------------+------+
| 伊泽瑞尔     |   24 |
+--------------+------+
1 row in set (0.01 sec)
```

//查询指定条件

删除表中数据

delete from table_name where 条件;

```
MariaDB [jsj]> delete from stu where age=24;
Query OK, 1 row affected (0.00 sec)

MariaDB [jsj]> select * from stu;
+-----------+------+
| name      | age  |
+-----------+------+
| 拉克丝    |   23 |
+-----------+------+
1 row in set (0.00 sec)
```

0x08. mysql 远程登陆

①root 密码设置

★ # mysql_secure_installation　安全安装(用于生产环境设置)

根据提示设置 root 密码

```
Change the root password? [Y/n]
```

输入 Y　进行设置密码(需要连续输入两次)

以后登陆 mysql 需要输入

★ # mysql -u root -p　//再输入密码

②远程登陆权限用户设置

语法: grant 权限 privileges on 数据库.表 to 用户@主机 identified by '密码';

例:grant select on test.info to stu@'%' identified by '123';

上面语句赋予 stu 用户在任何主机下登陆 select 查询 test 数据库下 info 表的查询权限,stu 密码为 123

最后

★ > flush privileges;　//刷新权限设置

★ # service mariadb restart;　//重新开启 mysql

③远程登陆

语句:# mysql -h 主机名/ip -u 用户名 -p

★ # mysql -h 192.168.60.104 -u stu -p

```
[root@localhost ~]# mysql -h 192.168.60.104 -u stu -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 129
Server version: 5.5.35-MariaDB MariaDB Server

Copyright (c) 2000, 2013, Oracle, Monty Program Ab and others.

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

MariaDB [(none)]>
```

④ 其他()

1.撤销权限

```
revoke privileges on *.* from username@host;
```

2. insert into 表 1 select * from 表 2

上面语句实现表 2 copy 到 表 1 的复制(需要两表结构相同)

3.用户信息表　mysql.user

```
MariaDB [(none)]> select user,password from mysql.user;
+------+-------------------------------------------+
| user | password                                  |
+------+-------------------------------------------+
| root | *66AAF2DC995EE0A8AB7A4EFC245E2ED87E4B7D39 |
| root | *66AAF2DC995EE0A8AB7A4EFC245E2ED87E4B7D39 |
| root | *66AAF2DC995EE0A8AB7A4EFC245E2ED87E4B7D39 |
| root | *66AAF2DC995EE0A8AB7A4EFC245E2ED87E4B7D39 |
| stu  | *23AE809DDACAF96AF0FD78ED04B6A265E05AA257 |
+------+-------------------------------------------+
5 rows in set (0.00 sec)

MariaDB [(none)]>
```

0x09. 使用 php 连接 Mysql

①安装 php 环境

★
```
# yum install php*     //安装
# service httpd restart    //重新启动 Apache
# setenforce 0             //关闭 SELinux 再次确认关闭
```

②后台代码

由于没开 PHP 课程下面程序可能不太好理解,可以先背下来

★
```
# vim stu.php
```

输入下方程序

```
<?php
$conn=mysql_connect('主机名/ip','用户名','密码');  //连接库
mysql_query('set names utf8');        //设置编码
if(!$conn)                  //判断是否连接成功
{
   echo "连接错误";
   exit();                //连接失败则退出程序
}
$query='select * from 表名';        //设置查询语句
$result=mysql_db_query('库名',$query);      //查询指定库
echo '<table border="1">';
while($row=mysql_fetch_array($result))
{
    echo   '<tr><td>'.$row[' 列 名  1'].'</td><td>'.$row[' 列 名
2'].'</td></tr>';           //循环输出查询结果
}
echo '</table>';
?>
```

ESC :wq 保存退出

访问 http://192.168.1.120/stuinfo.php //对应你的地址进行测试