Linux 01

今日学习目标

✅ Linux系统的认识以及部署安装
✅ 认识Linux文件系统结构
✅ 实际操作文件和文件夹等
✅ vi 编辑器的使用
✅ 文件传输
✅ host

学习进度

linux(5/12)

一天总结

学习 linux 记住一句话，“一切皆文件”，什么都好办了。

很多人一说到Linux，就会说自由。但是你真的而理解这种自由吗？这种自由是一种权力，能够决定你的计算机干什么，获取这种自由的唯一方式就是知道你的计算机在做什么。自由就是你的计算机没有任何秘密，你可以从计算机那里了解一切，只要你用心去寻找。

Linux 的作者 linus 是一个很牛逼的大神，昨天还看了2005年他在 Google 做讲座的视频，那时他在宣传 git （一个分布式的版本控制系统）,大神就是大神，整场都在 diss svn 以及 cvs 模式的版本控制工具，甚至还骂在坐的 Google 工程师蠢货...。因为他们问的问题要么太蠢，要么是质疑 git 为什么要采用分布式，习惯了使用集中式方式的他们简直被 linus 骂成了猪。黑客与画家，创造者等词联系在一起就不会那么难理解了吧。我们凡人是不懂得。。😟

黑客崇尚自由，这与 Linux 系统特别匹配，对这个系统你得负责，你得知道你在干什么。

扯那么多...开始正题，记录一下 linux 的学习过程。

环境说明：

- Macbook Pro 10.14版本
- VMware Fusion 10.1.3
- centos 6.5

环境搭建

1.安装 VMware Fusion略。

2.安装 centos 系统: 开头跳过检查（skip）。

3.配置网卡。

编辑网卡： vi /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
TYPE=Ethernet
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.235.100 #设置 IP 地址
NETMASK=255.255.255.0 #设置子网掩码
```

```
GATEWAY=192.168.235.2
DNS1=114.114.114.114
```

重启网卡：

`service network restart`

在这也可以配置DNS:

`vi /etc/resolv.conf`
nameserver 网关

4.关闭防火墙.

```
iptables -F
```

5.配置 host 以及hostname

host映射是在 /etc/hosts文件下进行配置：
初始是这样的，记录着本地回环地址与host的映射。

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
```

-

```
```

可以自己添加，比如

192.168.1.11 node1
192.168.1.12 node2

```
hostname是主机的标识，在 /etc/sysconfig/networking 中进行配置：
```

NETWORKING=yes
HOSTNAME=node01

认识 Linux文件系统

1.目录结构

首先得清楚 linux 系统都有那些约定俗成的目录，各自的目录都一般放什么文件。

bin: 存放二进制可执行文件（ls,cat,mkdir等）

sbin: 存放二进制文件，只有root用户才能访问

dev: 存放设备文件

usr: 存放系统应用程序，其中 /usr/local 是本地管理员软件安装目录

etc: 存放配置文件

lib: 存放一些共享类库和内核模块

tmp: 存放一些临时文件

var: 用于存放运行时需要改变数据的文件

boot: 存放开机所需的文件，开机时载入开机管理程序（bootloader），并映射到内存中。

home: 存放用户的东西

mnt:挂载目录

opt: 额外安装的可选软件的目录

文件的类型：

普通文件、目录文件、设备文件、链接文件、管道文件、套接字文件。

操作文件系统

孰能省巧，多练几遍

vi 编辑器

孰能省巧，多练几遍

host配置

配置 host 以及hostname

host映射是在 /etc/hosts文件下进行配置：
初始是这样的，记录着本地回环地址与host的映射。

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
```

```
# 可以自己添加，比如
192.168.1.11 node1
192.168.1.12 node2
```

hostname是主机的标识，在 /etc/sysconfig/networking 中进行配置：

```
NETWORKING=yes
HOSTNAME=node01
```

遇到的问题以及解决方案

问题1

【问题描述】

> mac 下如何保存ssh的连接信息,实现自动登录。

【问题思路】

在 windows 下有一款比较好用的 `xshell` 工具，但是 mac 下没有对应的替代品。但是有一款优秀的客户端 `iTerm2`。

【解决方案】

1.可以在【偏好设置】里面对应的【Profiles】页点击+号增加新的 profile 。

2.封装一个脚本，如下：

```
#!/usr/bin/expect -f
set timeout 60 # 设置超时时间
set username [lindex $argv 0] # 设置用户名
set password [lindex $argv 1] # 设置密码
set host [lindex $argv 2] # 设置ip或者域名
set port [lindex $argv 3] # 设置端口号
spawn ssh -p $port $username@$host # 启动一个的新的线程

# 模式 判断
expect {
        "*yes/no" { send "yes\r";exp_continue }
        "*password:" { send "$password\r" }
}
expect "$*" { send "pwd\r" }
interact
```

其中，Expect是一个用来处理**交互**的命令。借助Expect，我们可以将交互过程写在一个脚本上，使之自动化完成。提供了 send ，expect ，spawn ， interact 等命令。

- send：用于向进程发送字符串
- expect：从进程接收字符串
- spawn：启动新的进程
- interact：允许用户交互

说明：spawn 是用来启动一个进程的，在它之后的expect 和 send 都是与该进程交互的。interact 起到干预的作用。

3.在1步骤中的新建的 profile 中新增命令行的内容：

```
expect ~/.ssh/config_big_data/node01 root 123**4 cq.x***.te*h 4**2
```

因为上述连接涉及安全信息，故部分字符串被 * 代替。

依次为：expect 脚本 usernamepassword host port

4.总结，这种方式应该是比较有效的方式了，其实还可以把公钥传到服务器，采用秘钥的验证方式登录。

参考文章：

1.expect原理详解

2.expect脚本传参

问题2

【问题描述】

> 克隆虚拟机之网卡配置

【问题思路】

网卡的 mac 地址冲突。修改一下 mac 地址即可。

【解决方案】

方案1：

1. vi /etc/udev/rules.d/70-persistent-net.rules

注释掉eth0，然后将下面的eth1改名为eth0，复制下此eth0的mac地址

1. 将此 mac 地址跳到 ifcfg-eth0 中即可。
2. 重启服务器 `reboot -h now或者 intit 6`

方案2：

rm -rf /etc/udev/rules.d/70-persistent-net.rules
intit 0
修改网卡ip即可。

问题3

【问题描述】

> 服务器之间如何实现免密登录，以及原理

【问题思路】

SSH以**非对称加密**实现身份验证，所以它是比较安全的。

身份验证有多种途径，例如①其中一种方法是使用自动生成的公钥-私钥对来简单地加密网络连接，随后使用密码认证进行登录；②另一种方法是人工生成一对公钥和私钥，通过生成的密钥进行认证，这样就可以在不输入密码的情况下登录。任何人都可以自行生成密钥。公钥需要放在待访问的电脑之中，而对应的私钥需要由用户自行保管。认证过程基于生成出来的私钥，但整个认证过程私钥本身中不会传输到网络中。

【解决方案】

> 如何新建密钥对呢?

1.查是否已存在密钥对，打开终端（Terminal）：

输入：ls -al ~/.ssh

查看是否输出密钥对,如果有的话，会输出如下文件信息：id_rsa 和 id_rsa.pub

-rw------- 1 taoshilei staff 1679 Jun 25 22:34 id_rsa
-rw-r--r-- 1 taoshilei staff 403 Jun 25 22:34 id_rsa.pub

2、如果没有，则需要我们手动创建

输入：`ssh-keygen -t rsa -b 4096 -C "your_email"`

ssh-keygen 是生成秘钥的工具之一。

SSH supports several public key algorithms（公开秘钥算法） for authentication keys.

> 1.rsa – A key size of at least 2048 bits is recommended for RSA; 4096 bits is better.
> 2.dsa – DSA in its original form is no longer recommended.(不推荐使用)
> 3.ecdsa – Only three key sizes are supported: 256, 384, and 521 (sic!) bits.(大多数ssh客户端支持)
> 4.ed25519 – Support for it in clients is not yet universal. (还没有普及)

The algorithm is selected using the -t option and key size using the -b option.

-t 参数指定加密算法，-b 参数指定长度

用法如下：

```
ssh-keygen -t rsa -b 4096
```

3.将公钥发送到服务器

使用 ssh-copy-id 工具。

用法：

```
ssh-copy-id -i 公钥位置 user@host
```

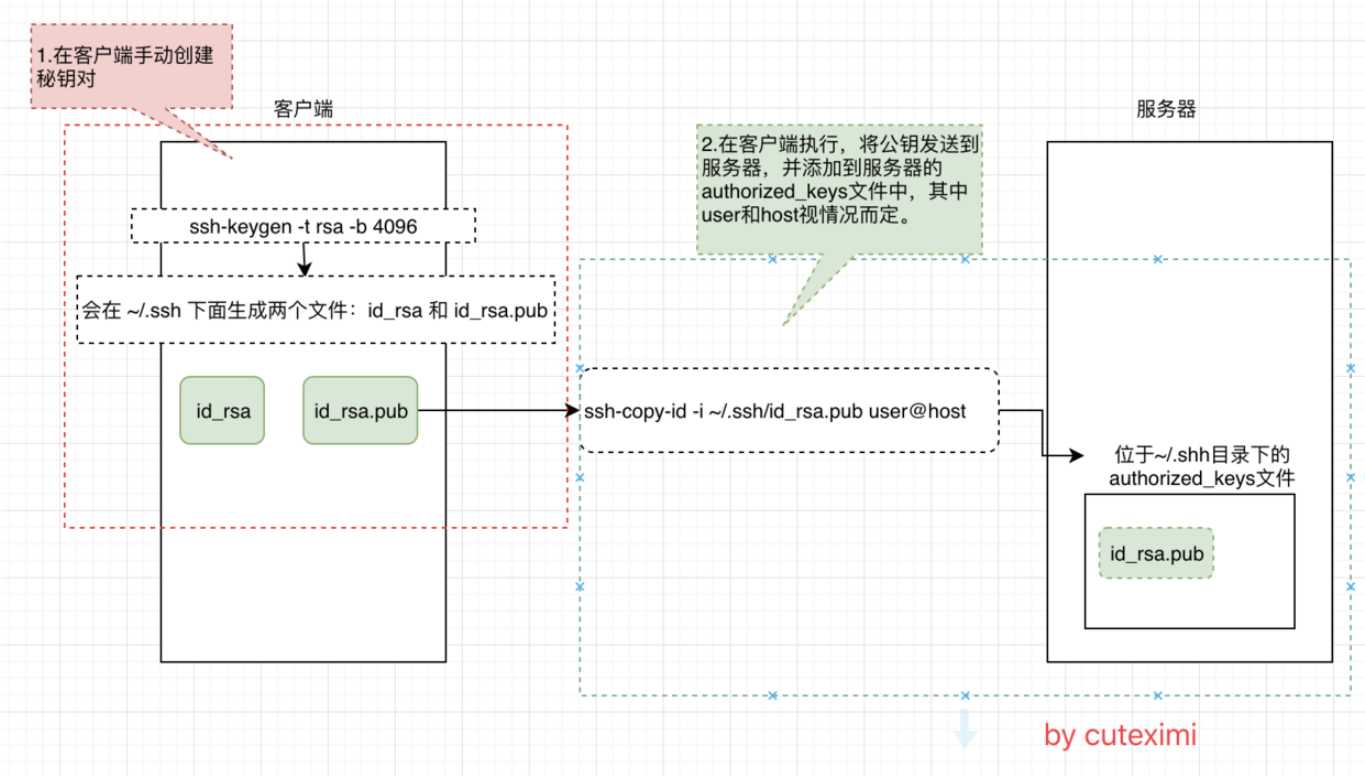
之后会提示输入密码进行认证。

在这之后，公钥就会被添加到 服务器上的 ~/.ssh/authorized_keys 文件了里面。

一旦在服务器上配置了公钥，服务器会允许任何具有私钥的客户端进行连接用户登录，在登录的过程中，客户端会通过数字签名交换来证明拥有私钥。

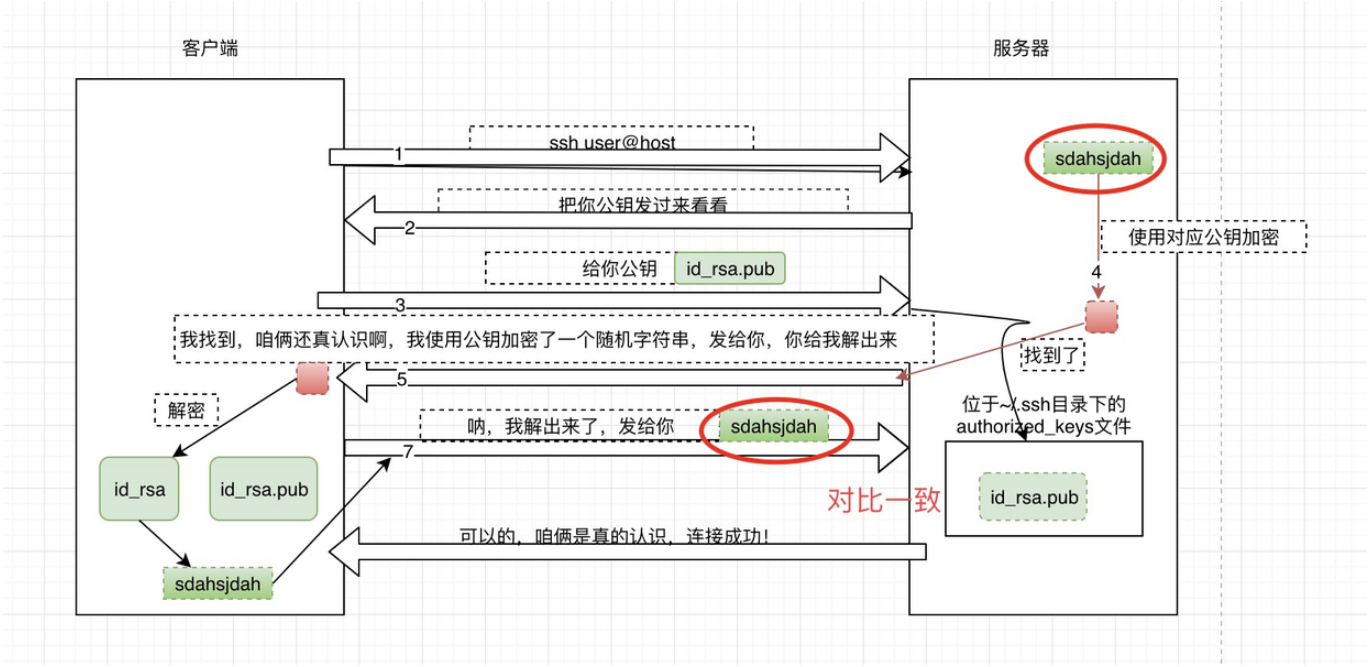
免密登录图解

创建密钥并添加到服务器上。



登录认证流程

值得注意的是：服务端的 .ssh目录权限必须是700（rwx------），authorized_keys文件的权限是600(rw-------)



部分参考

https://www.ssh.com/ssh/keygen/#sec-What-Is-ssh-keygen