1. 安装centos过程

工具：vmware 12， centos 6.5 minimal

安装过程出现的问题：

1. 、安装VMware后却不能安装64位虚拟机系统

解决：1、进入Bios，开机时不停地按F2键，进入Bios

2、选择advanced，找到CPU设置

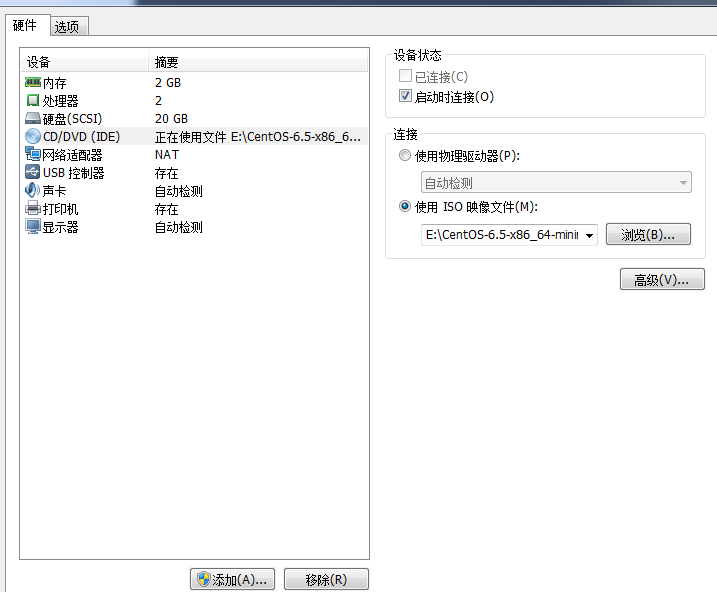
3、设置虚拟支持开启

②、安装centos 6.5 minimal 新建虚拟机的过程中（新建虚拟机的时候就选择了镜像），会突然卡死，安装不了

解决：新建虚拟机的时候选择稍后安装系统。



在虚拟机建好后选择系统镜像，然后开机。



安装过程中，系统提示是否test，选择skip。否则也会安装过程卡死。卡死后关机重启，重新选择skip。

1. SecureCRT连接虚拟机
2. 、配置网络

在VMware上安装CentOS-6.5 minimal - 配置网络CentOS的minimal版本默认不启动网络，所以安装完CentOS要自己配置网络。VMware虚拟机网络连接方式采用NAT方式（其他方式没试过）。

1 修改配置文件/etc/sysconfig/network-script/ifcfg-eth0

修改下面两项：

# 是否依赖Network Manager的控制，minimal没装这玩意，所以设为no

NM\_CONTROLLED="no"

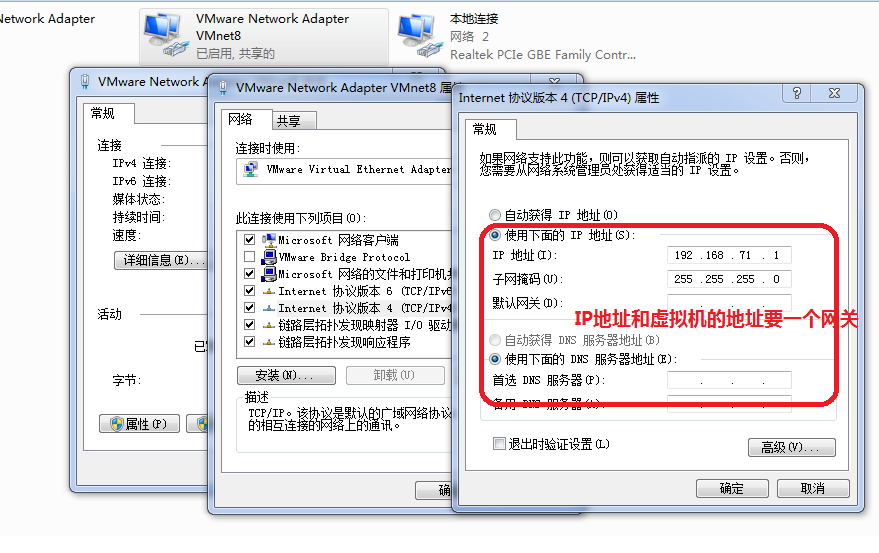
# 开机启动，当然yes

ONBOOT="yes"

2 重启服务

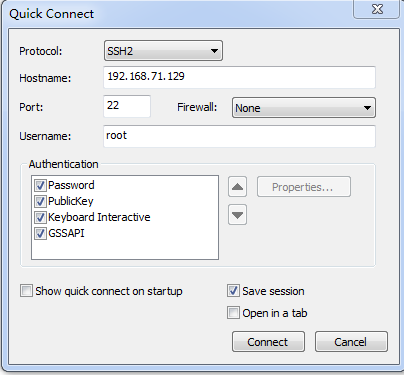
service network restart

1. 、SecureCRT连接
2. 配置windows7的虚拟网卡



1. SecureCRT

要保证虚拟机的linux系统安装了ssh并且开启了



1. 配置Apache httpd

<http://blog.csdn.net/u010551621/article/details/45046787>

httpd就是apache http server

#ps -e |grep httpd //查看是否有httpd进程

service httpd start

service httpd restart

service httpd stop

1. 安装git最新版本

获取源码 Wget <https://github.com/git/git/archive/master.zip>

yum install gcc

yum install curl

yum install curl-devel

yum install zlib-devel

yum install openssl-devel

yum install perl

yum install cpio

yum install expat-devel

yum install gettext-devel

#Can't locate ExtUtils/MakeMaker.pm in @INC 错误的解决方式

yum install perl-ExtUtils-CBuilder perl-ExtUtils-MakeMaker

# Git Make时出现：MSGFMT po/bg.msg make[1]: \*\*\* [po/bg.msg] 错误 127

yum install tcl build-essential tk gettext

# Git在make的时候报错：Can't locate ExtUtils/MakeMaker.pm in @INC

yum -y install perl-devel perl-CPAN

检查结果：git –version

1. 安装防火墙

1、安装iptables防火墙

#怎么知道系统是否安装了iptables?执行iptables -V，如果显示如：

iptables v1.4.7 #说明已经安装了iptables。

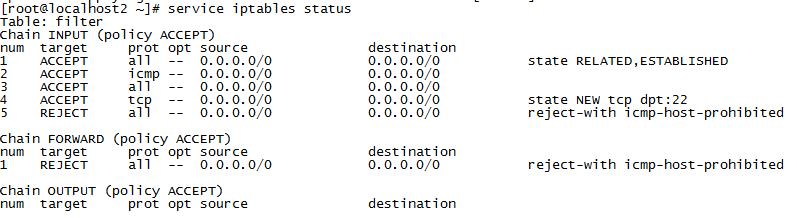
#如果没有安装iptables需要先安装，执行：

yum install iptables

service iptables start # stop、restart 开启、停止、重启

1. 防火墙的规则

防火墙的默认规则，policy ACCEPT标注，这表明当前链的默认策略为ACCEPT：



num – 指定链中的规则编号 ##一般是规则从前向后匹配，如果有匹配的了，就不看后面的规则了

target –特殊值：ACCEPT、DROP、QUEUE、RETURN

prot – 协议：tcp, udp, icmp等

source – 数据包的源IP地址

destination – 数据包的目标IP地址

iptables –F #清除防火墙的规则

service iptables save #防火墙的规则永久保存下来

#在tcp协议中，禁止所有的ip访问本机的1521端口。

iptables -I INPUT -p tcp --dport 80 -j DROP

#允许192.168.1.123访问本机的80端口，对应的应该是wm8的无线网卡的ip地址

iptables -I INPUT -s 192.168.71.1 -p tcp --dport 80 -j ACCEPT

#vmware的NAT模式对接的是wm8的虚拟网卡

#vmware的仅主机模式，是只能访问本机，不能连接外网，对接的vm1虚拟网卡

1. git使用总结

1、安装git的最新版本：获取源码wget <https://codeload.github.com/git/git/zip/master>

解压、编译安装./configure make make install

2、初始配置git

git config –global user.name “your name”

git config –global user.email “your email”

#在本地建立版本库

mkdir learngit

cd learngit

git init

3、在github上建立仓库

首先 注册并登陆你的 注册并登陆你的 注册并登陆你的 注册并登陆你的 注册并登陆你的 githubgithub github github账户，点击主页右侧的 账户，点击主页右侧的 账户，点击主页右侧的 账户，点击主页右侧的 账户，点击主页右侧的 +new repository+new repository +new repository+new repository+new repository +new repository+new repository +new repository+new repository+new repository按钮创建新的仓库 按钮创建新的仓库 按钮创建新的仓库 按钮创建新的仓库 。 输入仓库名称等内容并创建你的。 输入仓库名称等内容并创建你的。 输入仓库名称等内容并创建你的。 输入仓库名称等内容并创建你的。 输入仓库名称等内容并创建你的。 输入仓库名称等内容并创建你的。 输入仓库名称等内容并创建你的仓库。



创建分支可以：



1. 创建SSH Key

$ ssh-keygen -t rsa -C [youremail@example.com](mailto:youremail@example.com)

你需要把邮件地址换成你自己的邮件地址，然后一路回车，使用默认值即可，由于这个Key也不是用于军事目的，所以也无需设置密码。



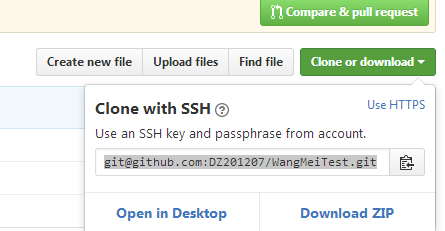
根据返回信息找到id\_rsa.pub文件

然后在**github**上添加**SSL Key。**登陆GitHub，打开“Account settings”，“SSH Keys”页面。然后，点击“Add SSH Key”，填上任意Title，在Key文本框里粘贴id\_rsa.pub文件的内容。



1. 从远程克隆仓库

Git clone git@github.com:DZ201207/WangMeiTest.git



$ git remote #要查看远程库的信息

$ git remote –v #用git remote -v显示更详细的信息

1. 关联远程库

git remote add origin git@github.com:michaelliao/learngit.git

#可以通过如下命令进行代码合并【注：pull=fetch+merge]

git pull --rebase origin master 或者 git pull origin master

git push -u origin master

1. 基本操作
2. 初始化一个Git仓库，使用git init命令。

添加文件到Git仓库，分两步：

第一步，使用命令git add <file>，注意，可反复多次使用，添加多个文件；

第二步，使用命令git commit，完成。

1. 版本回退

HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令git reset --hard commit\_id。

穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。

要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。

(3) 场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令git checkout -- file。

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令git reset HEAD file，就回到了场景1，第二步按场景1操作。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，版本回退一节，前提是没有推送到远程库。

1. 分支操作

查看分支：git branch

创建分支：git branch <name>

切换分支：git checkout <name>

创建+切换分支：git checkout -b <name>

合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>

强制删除没有合并分支：git branch -D <name>

1. 多人协作的工作模式通常是这样：

首先，可以试图用git push origin branch-name推送自己的修改；

如果推送失败，则因为远程分支比你的本地更新，需要先用git pull试图合并；

如果合并有冲突，则解决冲突，并在本地提交；

没有冲突或者解决掉冲突后，再用git push origin branch-name推送就能成功！

如果git pull提示“no tracking information”，则说明本地分支和远程分支的链接关系没有创建，用命令git branch --set-upstream branch-name origin/branch-name。

查看远程库信息，使用git remote -v；

本地新建的分支如果不推送到远程，对其他人就是不可见的；

从本地推送分支，使用git push origin branch-name，如果推送失败，先用git pull抓取远程的新提交；

在本地创建和远程分支对应的分支，使用git checkout -b branch-name origin/branch-name，本地和远程分支的名称最好一致；

建立本地分支和远程分支的关联，使用git branch --set-upstream branch-name origin/branch-name；

从远程抓取分支，使用git pull，如果有冲突，要先处理冲突。

1. 常见的命令

git checkout -b dev 相当于 git branch dev和git checkout dev

git checkout -b dev origin/dev #创建远程origin的dev分支到本地

git add read.txt

git commit –m “注释”

git push origin branch-name #把对应分支推送到远程

git pull #把最新的提交从origin/dev抓下来

git branch --set-upstream dev origin/dev #git pull失败的时候

git branch #查看本地分支

git branch –a #查看本地和远程的分支信息

git branch test #创建本地分支

git push origin test #把创建的本地分支推送到远程仓库

git branch –d test #删除本地分支

git push origin :test #删除远程分支

git checkout test #切换分支

git checkout -b <name> # 创建+切换分支

git checkout -- test.txt #用版本库里的版本替换工作区的版本，无论工作区是修改还是删除，都可以“一键还原”

git merge test01 #把分支test01合并到正在操作的分支

git log –graph #命令可以看到分支合并图

git status #查看文件状态

git diff #查看修改的地方

#版本回退

git log #查看每次提交的信息

git log –pretty=oneline #每条提交的信息，一行显示

git reset –hard id #id为版本号码

git reflog记录git的所以得操作信息

#

git checkout -- file可以丢弃工作区的修改

git reset HEAD file可以把暂存区的修改撤销掉（unstage），重新放回工作区

1. vim的基本操作

①、VIM的三种工作模式

命令模式

输入模式

行编辑模式

②、vim的启动和关闭

启动：vim filename

关闭： q:! 不存档强制退出

:w 保存但不退出

:wq 存档后退出

ZZ 保存退出

ZQ 不保存退出

③、编辑：新增 a：光标所在位置后面开始新增资料

A：从光标所在行最后面的地方开始新增资料

i ：从光标所在位置前面开始插入资料

I ：从光标所在行的第一个非空白字元前面开始插入资料。

o ：在光标所在行下新增一列并进入输入模式

O: 在光标所在行上方新增一列并进入输入模式

冒号进入行命令模式，输入set nu可设置vim显示行号

命令模式下的操作：删除与修改

x ： 删除光标所在字符

dw 删除一个单词

dd ：删除光标所在的行

s ：删除光标所在字节，并进入输入模式

#表示数字

#x 删除几个字符，如3x

#dw 删除几个单词，如3dw表示删除三个单词

#dd 删除多个行，如3dd 表示删除光标行及光标的下两行

d$ 删除光标到行尾的内容

nd上下光标键:

删除当前行之上、下的几行文本（包括当前行）

u:恢复刚才被修改的文本 U：恢复光标所在行的所有修改

.:重复上一次命令的操作

复制命令：y

yy: 复制整行

yw:复制光标所在的单词

nyw：复制包括光标所在的n个单词

nyy:复制包括当前行在内的n行

粘贴命令

p

光标移动

h、j、k、l

nG：移动到底n行行首

:/string 搜索指定的字符串。

按键n：继续进行搜索

如果写的是斜杠（/）：按n会从当前位置向后查找；如果写的是问号（？），按n键则从当前位置向前查找

查找且替换

:%s /SEARCH /REPLACE

把文档中所有SEARCH替换成REPLACE；

:#,# s /SEARCH /REPLACE /g

＃号表示数字，表示从多少行到多少行，把SEARCH替换成REPLACE；

:s /SEARCH /REPLACE /g

把当前光标所在行中的SEARCH单词，替换成REPLACE，并把所有SEARCH高亮显示

set nu ‘显示行号

set ruler ‘显示光标位置

set undolevel ‘显示光标位置

set smartindent ‘智能缩进

syntax on(或syn on) ‘语法高亮