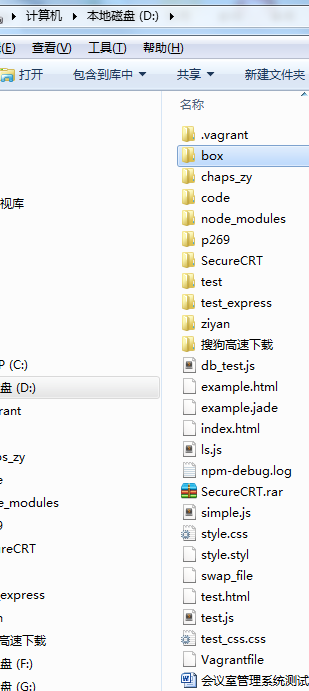
1. vagrant box add name url

eg：

vagrant box add debian32 ./box/ debian\_squeeze\_32.box



1. vagrant init 指定目录

eg: vagrant init debian32 ./

1. 修改配置文件

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

# Vagrantfile API/syntax version. Don't touch unless you know what you're doing!

VAGRANTFILE\_API\_VERSION = "2"

Vagrant.configure(VAGRANTFILE\_API\_VERSION) do |config|

# All Vagrant configuration is done here. The most common configuration

# options are documented and commented below. For a complete reference,

# please see the online documentation at vagrantup.com.

# Every Vagrant virtual environment requires a box to build off of.

config.vm.box = "debain"

# The url from where the 'config.vm.box' box will be fetched if it

# doesn't already exist on the user's system.

# config.vm.box\_url = "http://domain.com/path/to/above.box"

# Create a forwarded port mapping which allows access to a specific port

# within the machine from a port on the host machine. In the example below,

# accessing "localhost:8080" will access port 80 on the guest machine.

# config.vm.network :forwarded\_port, guest: 80, host: 8080

# Create a private network, which allows host-only access to the machine

# using a specific IP.

config.vm.network :private\_network, ip: "192.168.33.10"

# Create a public network, which generally matched to bridged network.

# Bridged networks make the machine appear as another physical device on

# your network.

# config.vm.network :public\_network

# If true, then any SSH connections made will enable agent forwarding.

# Default value: false

# config.ssh.forward\_agent = true

# Share an additional folder to the guest VM. The first argument is

# the path on the host to the actual folder. The second argument is

# the path on the guest to mount the folder. And the optional third

# argument is a set of non-required options.

config.vm.synced\_folder "./", "/d"

# Provider-specific configuration so you can fine-tune various

# backing providers for Vagrant. These expose provider-specific options.

# Example for VirtualBox:

#

# config.vm.provider :virtualbox do |vb|

# # Don't boot with headless mode

# vb.gui = true

#

# # Use VBoxManage to customize the VM. For example to change memory:

# vb.customize ["modifyvm", :id, "--memory", "1024"]

# end

#

# View the documentation for the provider you're using for more

# information on available options.

# Enable provisioning with Puppet stand alone. Puppet manifests

# are contained in a directory path relative to this Vagrantfile.

# You will need to create the manifests directory and a manifest in

# the file debain.pp in the manifests\_path directory.

#

# An example Puppet manifest to provision the message of the day:

#

# # group { "puppet":

# # ensure => "present",

# # }

# #

# # File { owner => 0, group => 0, mode => 0644 }

# #

# # file { '/etc/motd':

# # content => "Welcome to your Vagrant-built virtual machine!

# # Managed by Puppet.\n"

# # }

#

# config.vm.provision :puppet do |puppet|

# puppet.manifests\_path = "manifests"

# puppet.manifest\_file = "site.pp"

# end

# Enable provisioning with chef solo, specifying a cookbooks path, roles

# path, and data\_bags path (all relative to this Vagrantfile), and adding

# some recipes and/or roles.

#

# config.vm.provision :chef\_solo do |chef|

# chef.cookbooks\_path = "../my-recipes/cookbooks"

# chef.roles\_path = "../my-recipes/roles"

# chef.data\_bags\_path = "../my-recipes/data\_bags"

# chef.add\_recipe "mysql"

# chef.add\_role "web"

#

# # You may also specify custom JSON attributes:

# chef.json = { :mysql\_password => "foo" }

# end

# Enable provisioning with chef server, specifying the chef server URL,

# and the path to the validation key (relative to this Vagrantfile).

#

# The Opscode Platform uses HTTPS. Substitute your organization for

# ORGNAME in the URL and validation key.

#

# If you have your own Chef Server, use the appropriate URL, which may be

# HTTP instead of HTTPS depending on your configuration. Also change the

# validation key to validation.pem.

#

# config.vm.provision :chef\_client do |chef|

# chef.chef\_server\_url = "https://api.opscode.com/organizations/ORGNAME"

# chef.validation\_key\_path = "ORGNAME-validator.pem"

# end

#

# If you're using the Opscode platform, your validator client is

# ORGNAME-validator, replacing ORGNAME with your organization name.

#

# If you have your own Chef Server, the default validation client name is

# chef-validator, unless you changed the configuration.

#

# chef.validation\_client\_name = "ORGNAME-validator"

end

4.vagrant up

5.用户名 vagrant 密码 vagrant

6.sudo passwd

7.改用 root登录

8.copy syyx\_conf

9.设置 code- 软连接

10.安装mysql-client

11.安装php5-gd

12.更改配置文件里面的ip