**docker中centos7安装ssh服务**

原创 2017年03月22日 15:34:43

docker安装好后，自己动手做个自己的docker镜像，首先需要从服务器pull下一个基础的镜像，这里用centos7   
（最好是root用户下，我虚拟机中的docker普通用户运行docker提示服务没有运行，奇怪）   
一、下载官方原始镜像

docker pull centos:7

稍等一会下载完成，使用

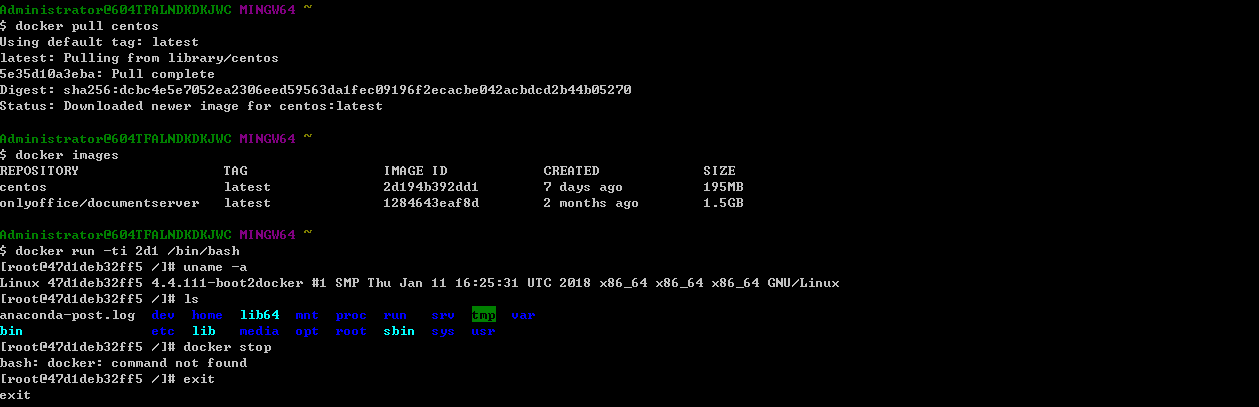
docker images

查看本地的镜像列表，如图，centos那个就是刚才下载的

  
接着用以下命令启动，使用image创建container并进入交互模式, login shell是/bin/bash

docker run -i -t centos:7 /bin/bash

接着终端就是容器中的centos的了，默认是root用户登录，接下来的操作都在容器中了



二、容器中安装passwd,openssl,openssh-server

yum install passwd openssl openssh-server -y

安装完成后   
启动sshd:

# /usr/sbin/sshd -D

这时报以下错误：   
[root@ b3426410ff43 /]# /usr/sbin/sshd   
Could not load host key: /etc/ssh/ssh\_host\_rsa\_key   
Could not load host key: /etc/ssh/ssh\_host\_ecdsa\_key   
Could not load host key: /etc/ssh/ssh\_host\_ed25519\_key

执行以下命令解决：

[root@b3426410ff43 /]# ssh-keygen -q -t rsa -b 2048 -f /etc/ssh/ssh\_host\_rsa\_key -N ''

[root@b3426410ff43 /]# ssh-keygen -q -t ecdsa -f /etc/ssh/ssh\_host\_ecdsa\_key -N ''

[root@b3426410ff43 /]# ssh-keygen -t dsa -f /etc/ssh/ssh\_host\_ed25519\_key -N ''

然后，修改 /etc/ssh/sshd\_config 配置信息：   
UsePAM yes 改为 UsePAM no   
UsePrivilegeSeparation sandbox 改为 UsePrivilegeSeparation no   
可以用vi改，也可以用下面命令

[root@b3426410ff43 /]# sed -i "s/#UsePrivilegeSeparation.\*/UsePrivilegeSeparation no/g" /etc/ssh/sshd\_config

[root@b3426410ff43 /]# sed -i "s/UsePAM.\*/UsePAM no/g" /etc/ssh/sshd\_config

修改完后，重新启动sshd

# /usr/sbin/sshd -D

[root@b1b202cf887c /]# passwd

Changing password for user root.

New password:

BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word

密码必须是 数字+字母+符号，位数大于8个

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

[root@b1b202cf887c /]# /usr/sbin/sshd -D

WARNING: 'UsePAM no' is not supported in Red Hat Enterprise Linux and may cause several problems.

[root@b3426410ff43 /]# /usr/sbin/sshd -D

然后修改root密码

passwd root

或者直接一句修改

echo “123456qxc$&″ | passwd --stdin root密码必须是 字母+数字+符号，位数大于8位

接着需要把修改后的镜像保存了，首先输入exit退出容器，再使用下面命令查看刚运行过的，

docker commit将修改后镜像保存到本地，参数是ID，名字

#docker ps -all

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

b3426410ff43 centos:7 "/bin/bash" 4 minutes ago Exited (0) 4 seconds ago centos7ssh

#docker commit b34 centos:v2

下次可以输入刚保存的名字启动修改过安装了ssh服务的镜像了。

**Docker启动一个Centos镜像**

接着上文，我们下载完成一个Centos镜像之后，开始启动

docker run -d -i -t <imageID> /bin/bash

这样就能启动一个一直停留在后台运行的Centos了。如果少了/bin/bash的话，Docker会生成一个Container但是马上就停止了，不会一致运行即使有了-d参数。

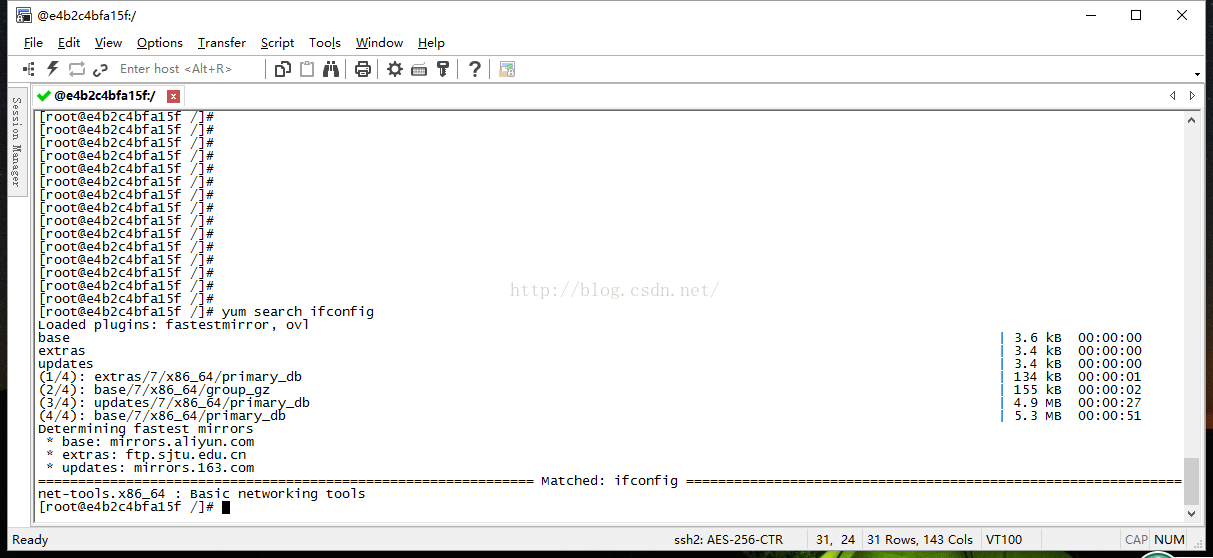
然后我们可以进入我们这个Centos

docker attach <ContainerID>

进入Centos后，发现竟然没有ifconfig和ssh，网上例子很多参照着来

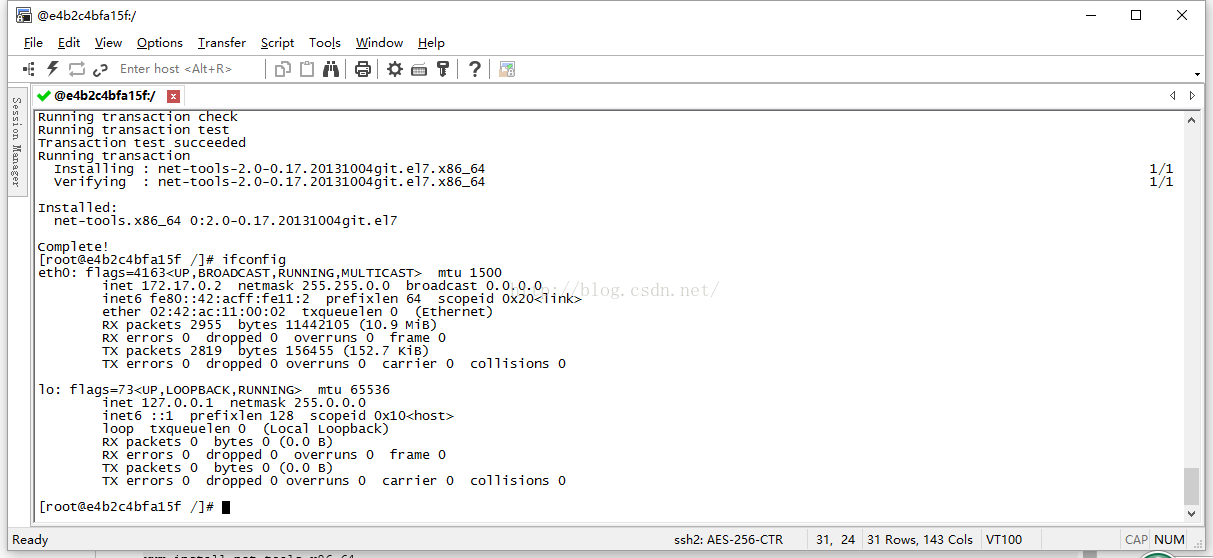
1. ifconfig的安装，执行命令

yum search ifconfig

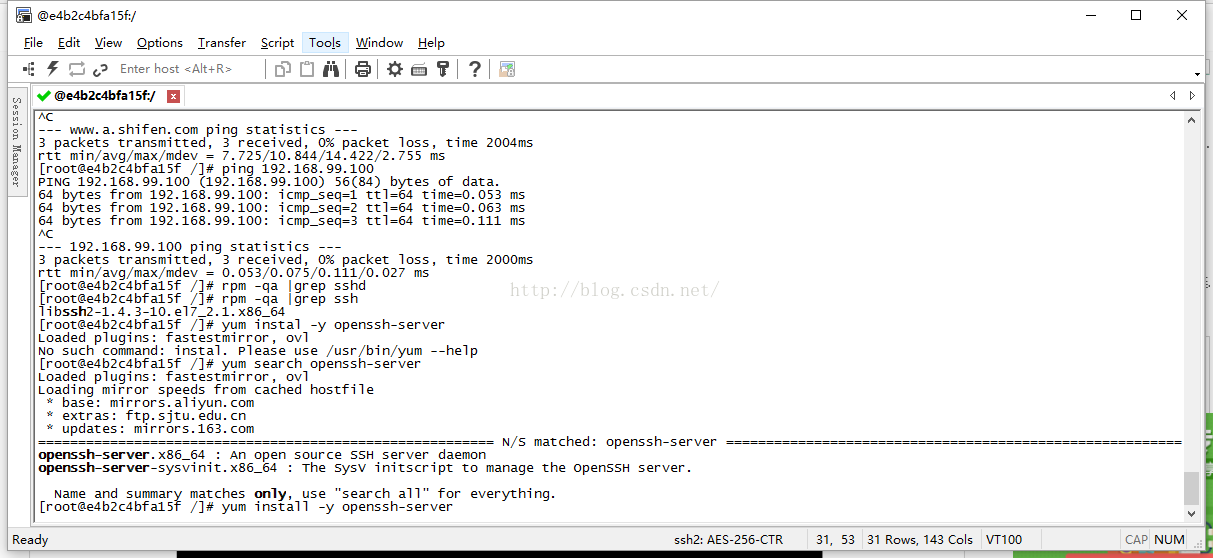


2. 安装ifconfig

yum install net-tools.x86\_64



好了，有了它终于可以知道自己的IP了和网络。现在还需要一个ssh让我们能方便的访问Centos

  
接下来只要启动就可以了

yum install -y openssh-server

#使用ssh-keygen命令来手动生成

ssh-keygen -q -t rsa -b 2048 -f /etc/ssh/ssh\_host\_rsa\_key -N ''

ssh-keygen -q -t ecdsa -f /etc/ssh/ssh\_host\_ecdsa\_key -N ''

ssh-keygen -t dsa -f /etc/ssh/ssh\_host\_ed25519\_key  -N ''

sed -i "s/#UsePrivilegeSeparation.\*/UsePrivilegeSeparation no/g" /etc/ssh/sshd\_config

sed -i "s/UsePAM.\*/UsePAM no/g" /etc/ssh/sshd\_config

启动ssh

/usr/sbin/sshd -D

接下来就是常用的命令了，将端口映射到宿主机，我这里就是VM分配的linux系统。

#退出，但不停止容器

Ctrl+P+Q

#回到Docker下面，停止容器

docker stop <容器ID>

#提交当前容器到镜像

docker commit <容器ID> <NAME:VERSION>

#启动新容器，并且进行端口映射

docker run -itd -p 50001:22 <刚才提交的镜像ID>  /bin/bash  
进入容器后再运行ssh

好了，这样我们在windows下利用ssh工具访问宿主机的IP端口就可以访问到容器了

我这里就是192.168.99.100:50001

$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

centos v4 65a70b3c749f 8 hours ago 544MB

centos v3 70f73f55aa5f 11 hours ago 386MB

centos v2 5b324eb3a8c4 11 hours ago 302MB

centos latest 2d194b392dd1 10 days ago 195MB

Administrator@604TFALNDKDKJWC MINGW64 ~

$ docker run -itd -p 50001:22 65a /bin/bash

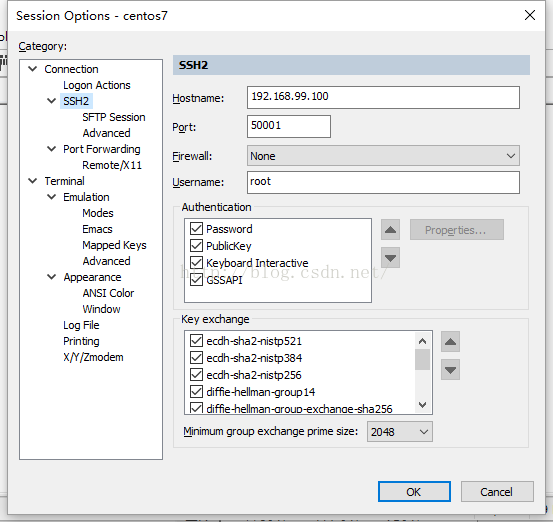
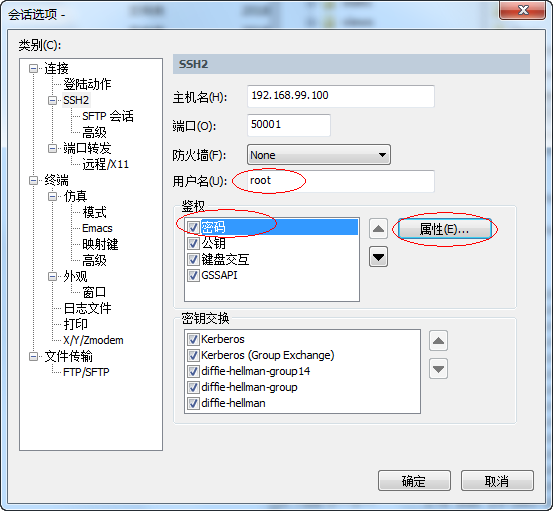
e044792450751d0ad3b93fac91d44c5fc03c59fc3e6c91baeda9845f50dd47d2

Administrator@604TFALNDKDKJWC MINGW64 ~

$ docker attach e04

[root@e04479245075 /]# /usr/sbin/sshd -D

WARNING: 'UsePAM no' is not supported in Red Hat Enterprise Linux and may cause several problems.



##### 2017-06-24 11:14

现在go 的跨平台编译比较简单了，

set GOARCH=amd64

set GOOS=linux

然后就可以了， go build出来的就是linux 64 位的可执行程序了

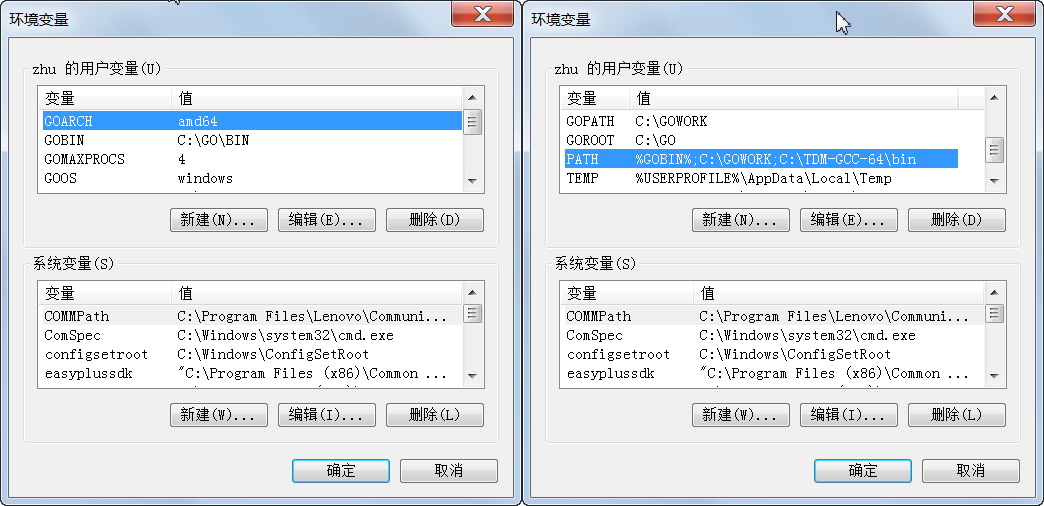
<https://www.cnblogs.com/lifeil/p/5408334.html>

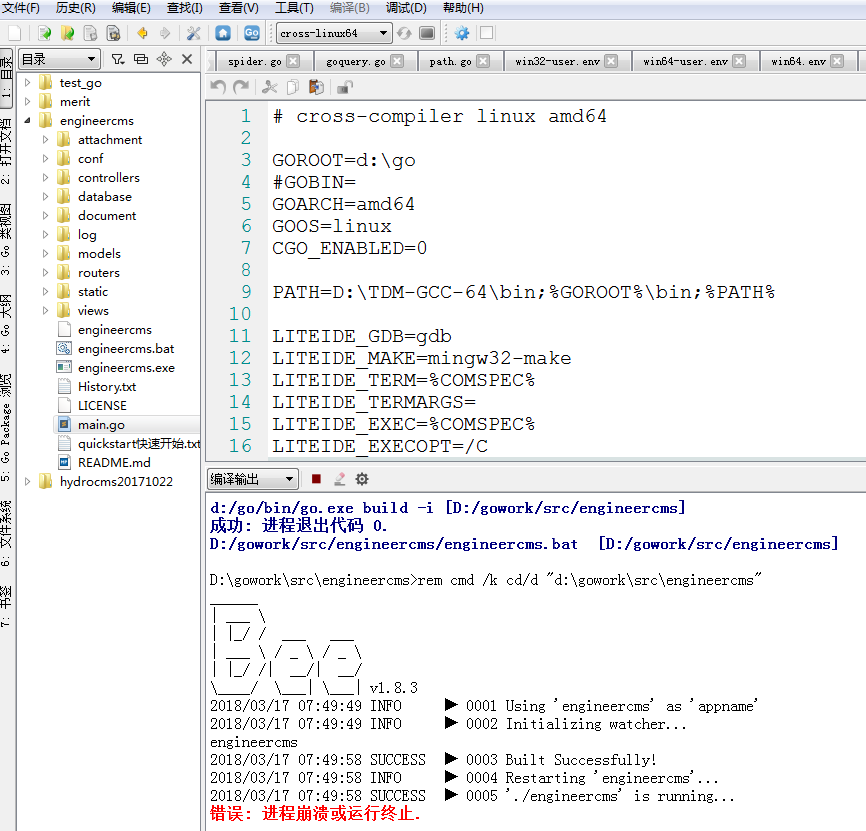
此时编译出来的可执行文件不带exe扩展名。对于Linux系统是没所谓的。

如果环境变量设置成以下图所示：

用LiteIDE编辑，会提示go install: cannot install cross-compiled binaries when GOBIN is set

所以，把GOBIN环境变量设置删除，然后重启LiteIDE（或重启机器？）就可以了。此时编译出来的可执行文件带exe扩展名。





上传到centos系统

输入ls -l列出当前目录下的文件。其中ls是list，也就是列出的意思，-l参数是long的意思，也就是列出文件详细信息，每行第一个字符带x的就是有可执行权限的文件，多半就是可执行程序。

使用./xxx来执行, 如果输入./xxx不能执行, 使用chmod +x filename来尝试给它执行的权限

[root@e04479245075 ~]# ./engineercms.exe

-bash: ./engineercms.exe: Permission denied

[root@e04479245075 ~]# chmod +x engineercms.exe

[root@e04479245075 ~]# ./engineercms.exe

[ORM]2018/03/16 23:48:38 register db `default`, sql: unknown driver "sqlite3" (forgotten import?)

must have one register DataBase alias named `default`

无法执行是因为应用中引用了基于cgo的sqlite3，而cgo不能跨平台，最好是在Linux系统中搭建环境进行编译。

