在VM中，设置虚拟网络编辑器 针对vmnet1和vmnet8设置相应的参数【之前一直存在一个误区就是，针对这vmnet1和vmnet8的ip必须要和本地对应，在同一个网段，其实并不是这样的】即使说本地IP地址是10.12.130.0，对于vmnet1和vmnet8子网IP，设置成192开头的也是可以的，当然需要注意的是vmnet1和vmnet8子网IP不能相同。

配置系统环境变量

对于Linux系统，环境变量的文件是profile 在etc目录下

使用命令 vim /etc/profile 进入配置配置文件 进行修改

以tomcat为例，如果想要在任何路径下 使用startup.sh 来启动tomcat服务，可以在配置文件进行如下配置

export CATALINA\_HOME=tomcat安装路径

export PATH=$PATH:$CATALINA\_HOME/bin

jdk:

export JAVA\_HOME=/usr/local/jdk1.8/jdk1.8.0\_181

export PATH=$PATH:$JAVA\_HOME/bin

export JRE\_HOME=$JAVA\_HOME/jre

export CLASSPATH=.:$JAVA\_HOME/lib:$JRE\_HOME/lib

查看防火墙 已经开放的端口

firewall-cmd --list-ports【使用firewalld来管理防火墙】

iptables –list-rules【使用iptables来管理防火墙】

windows下 启动服务

net start mysql

关闭防火墙

1） 永久性生效，重启后不会复原

开启： chkconfig iptables on

关闭： chkconfig iptables off

2） 即时生效，重启后复原

查看防火墙状态 service iptables status

开启： service iptables start

关闭： service iptables stop

^-^ 关闭firewalle 使用iptables 来管理防火墙

systemctl stop firewalld.service 关闭防火墙

在线下载

yum install iptables

yum install iptables-services

下载完成后 开启iptables

chkconfig iptables on | systemctl enable iptables.service【不建议关闭防火墙，如果需要放开某个端口，添加即可】

开放端口

vim /etc/sysconfig/iptables 添加如下语句

-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT

使用命令 service iptables restart 来更新修改后的配置文件

3) 配置虚拟机固定静态IP地址 方便远程连接

命令窗口输入： setup 设置参数

设置好 退出 重启网络服务 service network restart

4）添加一个用户

useradd 用户名

passwd 用户名 为该用户设置、修改密码

su - 用户名 切换到该用户，并将环境也切换到该用户的环境

id 查看当前用户

-----------------------------

Linux系统常用命令：

* VM两个Linux中进行文件传送

例如Linux-A IP地址为192.168.243.130，Linux-B IP地址为192.168.243.128

现在需要将A中 /home 目录下的temp.txt文件复制到B中/homee目录下，需要在B中执行以下命令

scp -r [root@192.168.243.130](mailto:root@192.168.243.130):/home/temp.txt /home【：后面不要有空格】

^-^ su - 切换到root用户

^-^卸载 rpm -e --nodeps 目标文件

^-^tail -10 文件名 显示最后10行该文件内容

grep 目标字段 文件名 显示在该文件中的目标字段

grep 目标字段 文件名 --color 高亮显示在该文件中的目标字段

^-^ ps -ef | grep java 查看名称中包括java的进程信息

最常用的方法是 ps aux | grep xxx 查找特定的进程 然后进行操作

^-^ netstat –anp | grep 8080 查看占用8080端口进程的信息

^-^ cat 文件名 在控制台打印该文件中的内容

查找一文件中某字段信息 cat 文件名 | grep 关键字

^-^ 根据文件名查找

Find / -name 文件名

^-^递归删除目标文件夹及其里面的文件夹或文件 rm -rf 目标文件夹

^-^unzip命令解压 unzip -oq example.zip -d example

在当前目录下，将example.zip解压到example目录中

^-^tar命令解压 tar -zxvf 目标压缩文件 -C 压缩目的路径

如： tar -zxvf aaa.tar.gz -C ../jdk 将当前路径下的aaa.tar.gz压缩到目录jdk中 jdk在aaa.tar.gz当前目录的上一级

^-^tar命令压缩Linux环境下的文件或文件名 tar -zcvf 目标路径/压缩名 目标压缩文件

如： tar -zcvf ./tmp/aaa.tar.gz ./aaa 在当前目录下的aaa文件夹压缩到当前tmp目录中，压缩文件名为aaa.tar.gz

^-^mv 重命名 mv aaa.a.a aaa 将aaa.a.a改名为aaa

^-^使用cp命令复制 如果要复制的是目录 需要在源目录前加上参数 -r 表示递归，如 cp -r aaa /urs/local/tomcat 将aaa目录复制到Tomcat目录下，如果说不是目录，省略参数-r即可

^-^查找一文件中某字段信息 cat 文件名 | grep 关键字

^-^sh startup.sh 启动Tomcat

^-^修改完配置文件后 需要使用source命令更新下 source 文件名

^-^pwd 打印当前路径

^-^既可以使用sh startup.sh 来启动 也可以使用 ./startup.sh

^-^剪切文件夹 mv 源文件夹 目的文件夹【其中源文件夹可以有多个 用空格隔开即可 当然该命令也适用于文件】

^-^重命名 rm 1.txt 2.txt 把1.txt文件 重命名为2.txt 当然如果不在同级目录，那么就是剪切并重命名

^-^删除文件夹 rm -rf 文件夹【也可同时删除多个】

^-^查看当前路径下所有文件夹的大小 du -ah --max-depth=1

^-^chkconfig --list 查看服务状态 chkconfig iptables off关闭防火墙 开机启动后状态也是不变的， 和 service iptables start不同

^-^如果在Tomcat的bin目录下 使用命令 startup.sh tail -f ../logs/catalina.out 可以以日志形式启动Tomcat

^-^往某文件中输入信息 echo 信息【需要用双引号】 目的文件

^-^ poweroff 关闭系统

reboot shutdown -r now 重启

^-^ 修改文件权限 chmod xxx 文件名

其中xxx 代表三个数字 7最高，表示具备所有权限 1最低，表示具备执行权限，另外第一个x代表属主权限，其次是属组权限，最后是其他用户权限。

如 chmod 755 a.txt 表示对该文件，属主的权限是7，可读可写可执行，属组权限为5 具备读和执行，其他用户权限为5 具备读和执行【r-4,w-2,x-1】

[root@ybhost bin]# ./startup.sh && tail -f ../logs/catalina.out

查看数据库连接是否成功

47.98.135.108(公)

172.16.164.48(私有)'

rsync /home/work/source.txtwork@192.168.0.10:/home/work/ #把本地的source.txt文件拷贝到192.168.0.10机器上的/home/work目录下

rsync work@192.168.0.10:/home/work/source.txt/home/work/ #把192.168.0.10机器上的source.txt文件拷贝到本地的/home/work目录下

rsync work@192.168.0.10:/home/work/source.txtwork@192.168.0.11:/home/work/ #把192.168.0.10机器上的source.txt文件拷贝到192.168.0.11机器的/home/work目录下

rsync -r /home/work/sourcedirwork@192.168.0.10:/home/work/ #拷贝文件夹，加-r参数

rsync -r /home/work/sourcedirwork@www.myhost.com:/home/work/ #使用主机名

rsync -r -v /home/work/sourcedirwork@www.myhost.com:/home/work/ #显示详情，加-v参数

rsync -r -v --exclude sourcedir/notinclude /home/work/sourcedir work@www.myhost.com:/home/work/ #排除子目录，注意：--exclude后面的路径不能为绝对路径，必须为相对路径才可以，否则匹配不上，就不会被排除掉。

Linux系统中，可以通过ctrl + insert来复制 shift + insert来粘贴

## Docker常用命令

Docker network ls 查看docker容器中的已有的网卡

Docker容器中自定义网卡

Docker network create -d overlay d1m\_overlay\_1【创建一个名为d1m\_overlay\_1的，类型为overlay的网卡】

配置docker国内镜像地址 使用命令：vi /etc/docker/daemon.json

创建该文件，并添加

{

"registry-mirrors": ["https://registry.docker.cn.com",<https://docker.mirrors.ustc.edu.cn>]

}

一个是docker中国区镜像，一个是中科大的docker镜像地址

* 安装docker

在centos中，使用命令 yum -y install docker-ce 或 yum -y install docker-ce.x86\_64

在 Ubuntu中，使用命令 wget -qO- <https://get.docker.com/> | sh

安装成功后 查看版本号 docker –v

docker安装在宿主机中的位置是 var/lib/docker

【如：在docker/containers目录下查看到在线拉取的镜像】

* **启动 停止Docker 【端口为9200 9300】**

**systemctl**命令是系统服务管理器指令，它是 service 和 chkconfig 两个命令组合。

l 启动docker：systemctl start docker

l 停止docker：systemctl stop docker

l 重启docker：systemctl restart docker

l 查看docker状态：systemctl status docker

l 开机启动：systemctl enable docker

* docker 下载镜像

命令：docker pull elasticsearch 【如果不指定版本，则会下载最新的，即latest】

* docker info 查看docker概要信息,可以用来测试服务是否启动了
* 查看主机上已经创建的容器

命令：docker ps -a 【-a表示包括停止运行的容器，不加的话，只显示运行中的】

* 查看docker主机上已有的docker镜像

命令：docker images

* 删除已经安装的容器

命令：docker rm 容器ID【ID可以写前几位即可】

* 删除已经安装的镜像

命令：docker rmi 镜像ID【可简写】

* 创建一个docker容器

docker run -d --name=my\_elk -p 9200:9200 -p 9300:9300 elasticsearch:5.6.11【通常

-d 是表示创建的容器会在后台运行，常用的有tomcat elasticsearch kibana 一般都会涉及到端口 -p 如果需要容器自动重启的话，需要加上参数 --restart=always】

* 如果需要创建的容器不涉及到端口之类的，即前台交互模式的，使用命令

docker run -id –name=容器名 镜像名：版本【不加版本则默认为最新，如果需要创建的容器对应的镜像没有下载，会先下载再创建容器】

启动成功后，必然可以通过 docker ps 来查看到对应的容器信息

* docker ps | grep el 【只要包含el关键字的容器都会显示出来，grep 命令是全面搜索正则表达式并打印出来，是一个强大的文本搜索工具】

需要的话，可以通过命令 docker logs -f 加上查询到的容器ID，查看容器启动的状态信息。

* 启动已经创建好的容器

docker start 容器ID 【快捷键 CTRL+INSERT 复制 SHIFT+INSERT 黏贴】

* 进入创建的容器中

docker exec -it 加上容器ID /bin/bash

* docker容器中进行文件的拷贝

docker cp 文件名 容器名 : 需要拷贝到的目录【从宿主拷贝到docker容器中】

docker cp 容器名 ：文件名 需要拷贝到的目录【从docker容器拷贝到宿主】

* docker安装nginx

docker run –id –name=my\_nginx –p 80:80 –v /宿主机目录【以方便操作来同步容器中的文件】 : /usr/share/nginx/html nginx:版本

* docker部署mysql数据库
  + 首先需要有mysql相应的镜像
  + 使用命令创建容器 docker run -id --name=my\_mysql -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=520025 mysql:版本【PS：3306：3306表示端口映射，前一个是指宿主机的端口号，后一个指mysql在容器docker中的端口号，前者是可以随意更改的，后者是固定的，这样使用连接数据库软件时，使用宿主机端口即可、-e表示具体的环境，指定了用户密码 用户名默认为root】
* 查看每个容器的IP地址

其实每个创建后的容器都是存在其自身固定的IP地址，同一个宿主环境下创建的多个容器之间是局域网的网络，互相是可以通信的

使用命令：docker inspect 容器名 or 容器ID

* 如果需要将已经有的镜像放到其他机器上使用，或给别人使用，可以备份然后迁移
  + 备份

docker save -o xxx.tar 镜像名

* + 迁移

docker load -I xxx.tar

* docker中安装elasticsearch容器

docker run -id –name=my\_els -p 9200:9200 -p 9300:9300 elasticsearch:版本【如果这里添加了文件映射，虽然可以创建成功，但无法启动，至于原因，未解决】

如果要使用elasticsearch-head来连接会出现跨域的问题，需要进入容器中，在elasticsearch.yml文件中添加 http:cors.enabled: true http.cors.allow-origin: “\*”

重启容器即可

【docker部署redis后，如果需要进入客户端，使用命令docker exec -it 容器名 redis-cli】

* 在docker容器中安装命令 如vim

初始时下载地址是海外地址，所以速度很慢，可以先执行一下命令

mv /etc/apt/sources.list /etc/apt/sources.list.bak

echo "deb http://mirrors.163.com/debian/ jessie main non-free contrib" >> /etc/apt/sources.list

echo "deb http://mirrors.163.com/debian/ jessie-proposed-updates main non-free contrib" >>/etc/apt/sources.list

echo "deb-src http://mirrors.163.com/debian/ jessie main non-free contrib" >>/etc/apt/sources.list

echo "deb-src http://mirrors.163.com/debian/ jessie-proposed-updates main non-free contrib" >>/etc/apt/sources.list

执行完后，依次执行 apt-get update apt-get install vim

ls -a 查看当前目录下所有文件

* 创建镜像私库

STEP ONE : pull private repository image registry:2.6.2

Command: docker pull registry:2.6.2

STEP TWO: create and start a private repository docker

Command: docker run -id –name=registry –restart always -p 5000:5000 registry:2.6.2

STEP THREE: input url such as 192.168.243.128:5000/v2/\_catalog, if you see some information on the screen like {“repository”:[]} that means private warehouse was created successfully and the content is empty

STEP FOUR: alter file named daemon.json if exist [the file root is /etc/docker/daemon.json]

Add information:

{

"registry-mirrors": ["https://docker.mirrors.ustc.edu.cn"]

,"insecure-registries":["192.168.40.128:5000"]

}

STEP FIVE: restart docker and private warehouse

Command: systemctl daemon-reload

systemctl restart docker

systemctl start registry

STEP SIX:

First use a image which you have created already before to tag this image as one of the private warehouse’s image

Command: docker tag my\_redis:4.0.11 192.168.243.128:5000/my\_redis:4.0.11

If you haven’t start your private warehouse yet, you need to use command with “docker start registry” to start your warehouse

Then push your image which has been tagged before

Command: docker push 192.168.243.128:5000/my\_redis:4.0.11[If you haven’t modified the daemon.json file yet, there will be new problem mentioned: server gave HTTP response to HTTPS client, "insecure-registries":["192.168.40.128:5000"] is needed, of course, there is also other way to handle this problem]

Wait for a second, then refresh your browser you will see something new on the screen

* 在私库拉取上传的镜像

Normally, you just need to use a command “docker pull 192.168.243.128:5000/my\_redis:4.0.11” to pull the image, but if you failed, maybe you need to configure some information in a file named daemon.json [It’s been mentioned before, /usr/docker/daemon.json] The details are as follows:

{

"registry-mirrors": ["https://docker.mirrors.ustc.edu.cn"]

,"insecure-registries":["192.168.40.128:5000"]

}

If necessary, add https before url

That’s because xshell client is a security terminal simulation software based ssh

So if you use one Linux system to pull images from other Linux system ipv4 address, you must add https before ipv4 like <https://192.168.243.128:5000>

## 查看日志

d ps 查看对应docker的ID 【docker ps 的简写】

根据ID来查看对应的日志 dt ID