**确保虚拟机能连外网**

1.在hyper-v上安装系统为 centos7的虚拟机，将网络方式配置为 内部网络（默认会新建1个名字为DefaultSwitch 的 虚拟交换机）

2.打开物理机器的控制面板\网络和 Internet\网络连接 ，找到上一步的虚拟交换机，右键属性，选择 IPv4协议，将ip地址设置为 192.168.125.1 （此ip地址需要和下一步中的虚拟机的网络配置一致）， 子网掩码设置为 255.255.255.0

3.修改配置，开启虚拟机的网络，固定虚拟机的ip地址

进入/etc/sysconfig/network-scripts 目录

使用vi 打开 ifcfg-eth0 这个文件

修改以下几列

BOOTPROTO=”static” //表示为固定ip

ONBOOT=”yes” //表示开机自动启动

IPADDR=192.168.125.200 //以下几列需要和上一步的虚拟交换机的配置对应

GATEWAY=192.168.125.1

DNS1=192.168.125.1

NETMASK=255.255.255.0

修改完成后 使用 systemctl restart network 进行生效

使用ifconfig命令确认是否生效

可以在物理机上ping 192.168.125.200 来确认是否可以连上虚拟机

4.使用终端工具来连接虚拟机（如mobaxterm）,需先确定虚拟机22端口没有开通防火墙

5.在虚拟机中运行 ping [www.baidu.com](http://www.baidu.com) 和 wget <http://www.baidu.com> 命令来确定虚拟机可以访问外网，如果访问不了外网，则在控制面板->网络和共享中心->更改适配器设置中，选中此时物理机使用的网卡，右键属性点击共享选项， ”复选框选中“在允许其他网络用。。。”并在下拉框中选择对应的虚拟路由器

**Docker简介**

一种虚拟化技术，基于linux的容器 （LXC）技术，对其进行了封装

比虚拟机要轻量，占用资源较小，基于进程，容器的环境依赖于宿主操作系统

快速高效的搭建环境，确保运行环境的统一

**Docker的安装**

1.uname -r ， 只有3.10以上的版本才可以安装

2.sudo yum update

3. 如果之前有安装docker,需要先卸载

sudo yum remove docker \

docker-client \

docker-client-latest \

docker-common \

docker-latest \

docker-latest-logrotate \

docker-logrotate \

docker-engine

4.安装依赖和添加yum源

sudo yum install -y yum-utils

sudo yum-config-manager \

--add-repo \

https://download.docker.com/linux/centos/docker-ce.repo

5.查看可以下载的Docker版本，

yum list docker-ce --showduplicates | sort -r

安装最新的稳定的版本

sudo yum install docker-ce docker-ce-cli containerd.io

如果不想安装最新版本，则可以指定安装的版本

sudo yum install docker-ce-<这里填版本> docker-ce-cli-<这里填版本> containerd.io

6.启动 docker 并设置为开机启动

sudo systemctl start docker

sudo systemctl enable docker

7.验证docker是否安装成功

docker version

**Docker的安装和启动**

查看当前安装的版本

docker -v

修改镜像的下载源（通过修改某个配置文件）

vi /etc/docker/daemon.json

在其中添加如下内容 （阿里云的镜像）

{

“registry-mirrors”:["https://bjtzu1jb.mirror.aliyuncs.com"]

}

systemctl daemon-reload && systemctl restart docker

可以增加下载速度

启动和停止docker 服务，及查看状态

systemctl start docker

systemctl stop docker

systemctl status docker

开机自动启动

system enable docker

**Docker镜像相关的命令**

查看镜像

docker images

搜索镜像 (搜索的是网络上的镜像)

docker search 镜像名称

拉取镜像

docker pull 镜像名称

删除镜像

docker rmi 镜像名称/镜像id

**Docker容器相关的命令**

查看运行中的容器

docker ps

如果是查看所有状态的容器，则为 docker ps -a

创建容器

docker run

例如：安装和运行第一个容器

docker pull hello-world

docker run hello-world

对应的参数

-i 表示运行容器

-t 表示容器启动后会进入命令行，即容器建立后会登录进去，分配一个伪终端

-v 表示目录映射关系，（前为宿主目录，后为容器内目录），这样在宿主机的修改会体现在容器内

-d 建立一个守护进程在后台运行 （-t 和 -d 二选一）

-p 表示端口映射， （前为宿主机端口，后为容器端口）

-name 表示为创建的容器起个名字

例子：docker run -it --name=mycentos centos:latest /bin/bash

通过交互方式建立容器，以centos:latest 镜像作为基础，取名mycentos，启动后执行镜像的/bin/bash 命令行

通过 exit 退出容器

例子：docker run -id --name=mycentos1 centos:latest

通过守护式（后台运行）方式创建容器

通过docker exec -it mycentos1 /bin/bash 来进入到某个正在运行的镜像中的命令行

例子：在启动容器时，实现目录挂载(/usr/test 映射到容器内的/usr/test)

docker run -id -v /usr/test:/usr/test --name=mycentos1 centos:latest

删除镜像 ，可以通过id 或者name 来删除

docker rmi hello-world

启动 停止 删除 容器，

docker start 容器id

docker stop 容器id

docker rm 容器id

复制文件到容器内部

docker cp 宿主机需要复制的文件路径 容器名称:容器路径

例如： docker cp /usr/5.txt mycentos:/us

将容器内的文件copy出来

docker cp 容器名称:容器路径 宿主机需要复制的文件路径

查看容器信息（包括ip地址等）

docker inspect 容器名称/id

参考： https://www.cnblogs.com/simawenbo/p/13644326.html

<https://www.cnblogs.com/simawenbo/p/13652454.html>

**Docker中安装mysql**

1. 拉取mysql 镜像

docker pull mysql

1. 创建容器

docker run -p 3306:3306 --name mysql -e MYSQL\_ROOT\_PASSWORD=123456 \

-v /usr/local/docker/mysql/conf/my.cnf:/etc/mysql/my.cnf \

-v /usr/local/docker/mysql/logs:/var/log/mysql \

-v /usr/local/docker/mysql/data:/var/lib/mysql \

-d mysql --lower\_case\_table\_names=1

（其中 -p 映射端口号

-e 设置mysql root用户密码

-name 设置容器名称

-v 把容器内部文件挂载至宿主机中，方便修改。

--lower\_case\_table\_names=1 忽略表名大小写，存储都用小写，不加可能无法正常启动）

3.执行以上命令， 会出现报错，原因是my.cnf没有配置,需要进行如下操作

删除宿主机 /usr/local/docker/mysql/conf/ 中的 my.cnf 文件夹，并在 /usr/local/src/mysql/conf/ 新建my.cnf文件，文件内容如下：

[client]

default-character-set=utf8mb4

[mysql]

default-character-set=utf8mb4

[mysqld]

pid-file = /var/run/mysqld/mysqld.pid

socket = /var/run/mysqld/mysqld.sock

datadir = /var/lib/mysql

secure-file-priv= NULL

# Disabling symbolic-links is recommended to prevent assorted security risks

symbolic-links=0

max\_connections=10000

default-time\_zone='+8:00'

character-set-client-handshake=FALSE

character\_set\_server=utf8mb4

collation-server=utf8mb4\_unicode\_ci

init\_connect='SET NAMES utf8mb4 COLLATE utf8mb4\_unicode\_ci'

# Custom config should go here

!includedir /etc/mysql/conf.d/

1. 启动容器

docker start mysql

可以通过之前介绍的docker images 和docker ps 命令 来查看docker运行情况

1. 进入容器，通过 mysql -uroot -p 命令并输入密码登录mysql

配置mysql

新建用户 xl 设置密码为 1010 ,被赋予其远程访问权限

CREATE USER 'xl'@'%' IDENTIFIED WITH mysql\_native\_password BY '1010';

GRANT ALL PRIVILEGES ON \*.\* TO 'xl'@'%';

1. 通过mysql 客户端 连接 虚拟机的3306端口，并使用指定的账号登录

参考： https://www.cnblogs.com/simawenbo/p/13644326.html

https://www.cnblogs.com/simawenbo/p/13652454.html

**Docker中安装nginx**

1.拉取nginx镜像

docker pull nginx

2.创建容器

docker run -id --name=mynginx -p 80:80 nginx

通过在浏览器中输入宿主机的ip来访问

1. 进入容器，对nginx进行配置

docker exec -it mynginx /bin/bash

nginx的安装目录在 /etc/nginx 下

可以通过读取nginx.conf 文件进行配置

也可以通过在/etc/nginx/conf.d 目录下的配置文件来进行模块化管理

通过/etc/nginx/conf.d/default.conf 文件可知，默认的静态文件根目录为/usr/share/nginx/html/

修改其内的index.html 来改变nginx首页

**Docker中安装redis**

1. 拉取redis镜像

docker pull redis:latest

1. 创建容器

docker run -id --name redis-test -p 6379:6379 redis

（其中 -p 映射端口号

-name 设置容器名称发 ）

1. 启动容器 docker start redis-test

4.通过客户端工具(RedisView)连接redis

5.在客户端工具中运行以下命令来设置redis密码

6. config set requirepass “你的密码”

**迁移和备份**

1.将容器保存为镜像

docker commit mynginx mynginx-image

然后可以通过此镜像再建立容器

例如：docker run -id --name mynginx-image1 -p 81:80 mynginx-image

提供宿主机的81端口来访问

2.镜像的备份和回复

备份 docker save -o mynginx.tar mynginx-image

恢复 docker load -i mynginx.tar （首先要删除 对应的容器和镜像）

**Dockerfile**

1. dockerfile 是一个由一系列命令组成的脚本，用于基于一个基础镜像(例如centos ,ubuntu)建立一个新的镜像

可以更方便的构造一个新的镜像

1. 常用命令示例

FROM nginx （From : 基于镜像）

RUN echo '这是一个本地构建的nginx镜像' > /usr/share/nginx/html/index.html （Run: 执行命令）

ADD /usr/local /usr/local (ADD 将宿主机的文件复制到容器内，如果是压缩文件，则解压

Env key value (Env 设置环境变量)

WORKDIR path (WORKDIR 设置工作目录)

MAINTAINER username (MAINTAINER 声明镜像的创建者)

//通过dockfile文件建立镜像

docker build -t=’imagename’ .

3.基于centos 基础镜像 建立 jdk 1.8镜像

**Docker 私有仓库**

1.搭建私有仓库

2.将镜像上传到私有仓库

3.在其他机器上下载镜像