Docker镜像服务部署帮助文档

武汉微创光电股份有限公司

2018年8月6日

版本历史信息

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 时间 | 作者 | 修改主要原因 |
| 1.0 | 2018-8-6 | 段文杰 | 文档创建 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1：项目介绍

帮助管理人员部署和配置socker镜像服务

2:软件安装列表

|  |  |
| --- | --- |
| 软件 | 说明 |
| CentOS-6.5-x86\_64-bin-DVD1.iso | 64位操作系统安装文件 |
| VMware-workstation-full-10.0.0-1295980.exe | 虚拟机客户端安装文件 |
| docker1.7.1离线安装依赖包及自动安装脚本 | device-mapper-libs-1.02.117-12.el6.x86\_64.rpm  docker-io-1.7.1-2.el6.x86\_64.rpm  libcgroup-0.40.rc1-23.el6.x86\_64.rpm  lua-alt-getopt-0.7.0-1.el6.noarch.rpm  lua-filesystem-1.4.2-1.el6.x86\_64.rpm  lua-lxc-1.0.9-1.el6.x86\_64.rpm  lxc-1.0.9-1.el6.x86\_64.rpm  lxc-libs-1.0.9-1.el6.x86\_64.rpm |

3 操作系统安装

3.1准备软件列表

|  |  |
| --- | --- |
| 软件 | 说明 |
| CentOS-6.5-x86\_64-bin-DVD1.iso | 64位操作系统安装文件 |
| VMware-workstation-full-10.0.0-1295980.exe | 虚拟机客户端安装文件 |

3.2详细启动歩骤

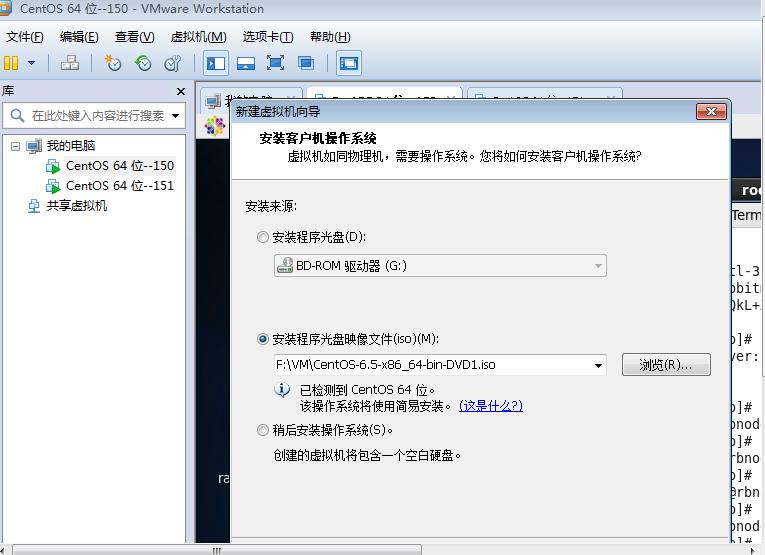
【sept1】打开新建虚拟机



【sept2】安装程序映像光盘iso



【sept3】选中iso文件



【sept4】简易安装信息



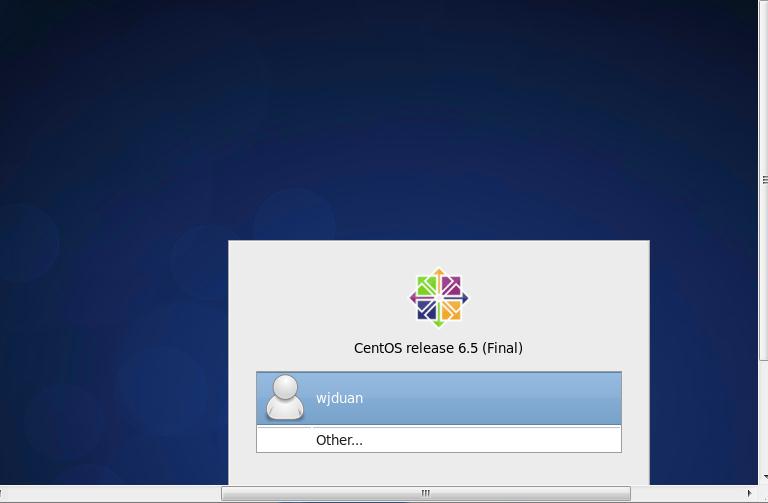
【sept5】指定安装位置



【sept6】安装过程中

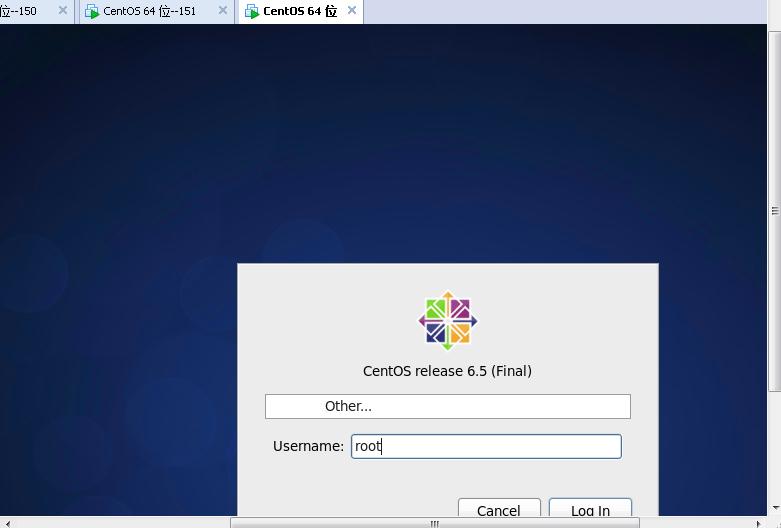


【sept7】安装完成

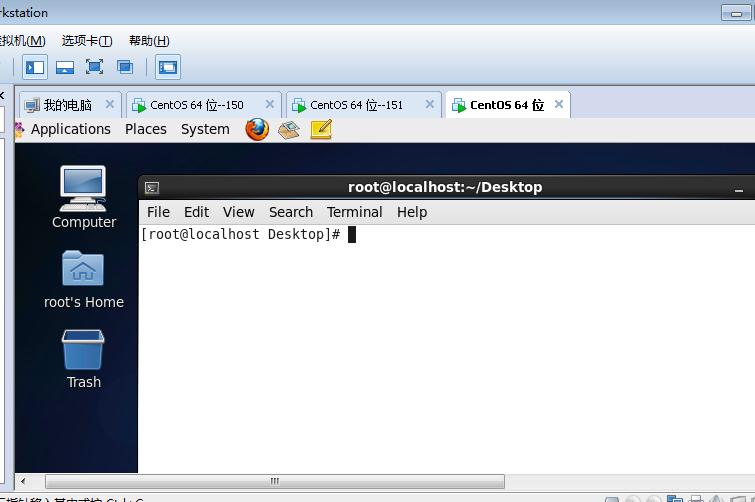


【sept8】点击other

使用root登录 密码：111111



【sept9】登录成功



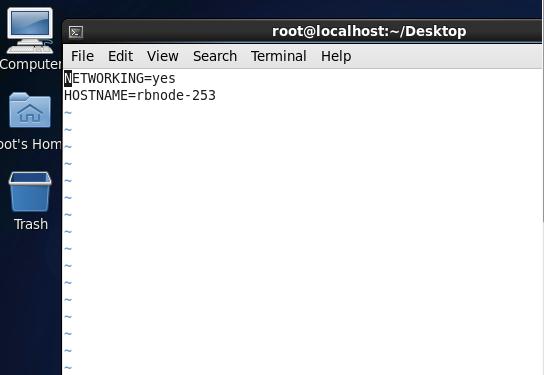
4：linux下静态IP设定

4.1：linux下设置静态IP

【sept1】修改主机名称

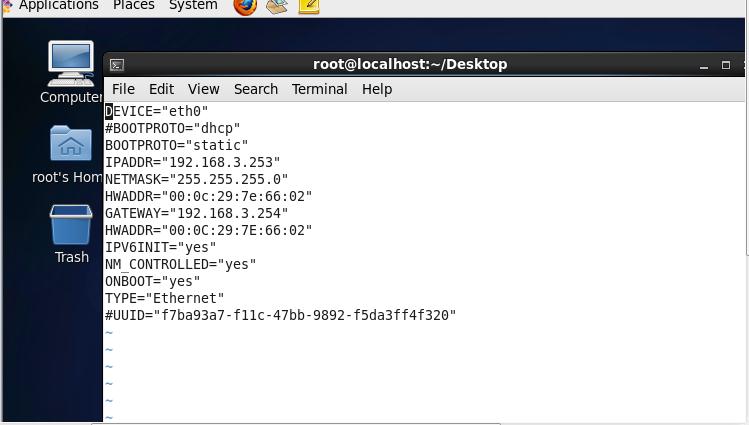
hostname=rbnode-253

命令：vi /etc/sysconfig/network



【sept2】配置静态ip地址

Vi /etc/sysconfig/network-scripts/ifcfg-eth0



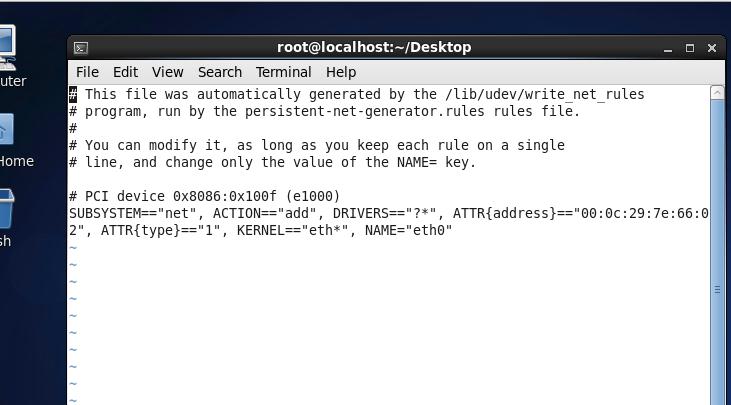
4.2：liunx下mac报错导致问题问题

Error:No suitable device found :no device found for connection ‘system eth0’

解决方案：由于网络的mac地址不同导致，修改mac即可

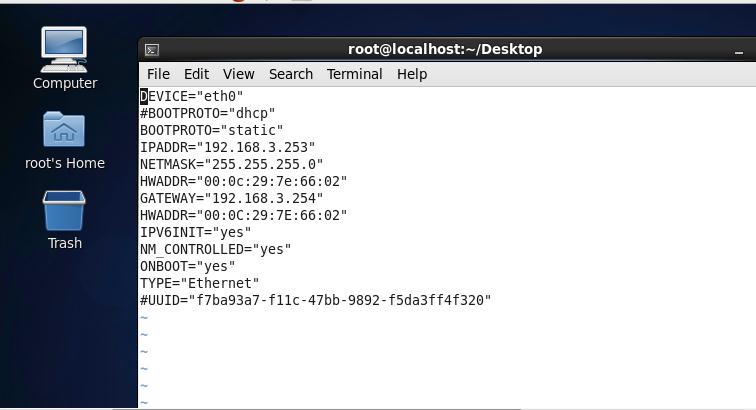
vim /etc/udev/rules.d/70-persistent-net.rules 找到里面的Attr{address}==”00:0c

:29:7e:66:02”



拷贝mac地址到ifcfg-eth0中

Vi /ect/sysconfig/network-script/ifcfg.eth0



测试是否链通ping 192.168.3.76

5：安装docker服务

5.1准备软件列表；

【sept0】文件存放路径:

[\\192.168.3.3\pis-publish\项目开发环境\Java\Java后台服务组件\docker内网部署及服务组件](\\\\192.168.3.3\\pis-publish\\项目开发环境\\Java\\Java后台服务组件\\docker内网部署及服务组件)

前提：拷贝软件到linux /home/tools/目录下;tools是本人新创建的；

|  |  |
| --- | --- |
| 软件 | 说明 |
| docker1.7.1离线安装依赖包及自动安装脚本 | device-mapper-libs-1.02.117-12.el6.x86\_64.rpm  docker-io-1.7.1-2.el6.x86\_64.rpm  libcgroup-0.40.rc1-23.el6.x86\_64.rpm  lua-alt-getopt-0.7.0-1.el6.noarch.rpm  lua-filesystem-1.4.2-1.el6.x86\_64.rpm  lua-lxc-1.0.9-1.el6.x86\_64.rpm  lxc-1.0.9-1.el6.x86\_64.rpm  lxc-libs-1.0.9-1.el6.x86\_64.rpm |

将tar包下的文件拷贝到/home/tools/loadMenu/下：

|  |  |
| --- | --- |
| 软件 | 说明 |
| tar包 | centos-nginx-gridfs.tar  mongo3.0.tar  mysql-jwzh.tar  nginx1.10.tar  rabbitmq-3.6.1-management.tar  redis3.0.tar  registry5000.tar  tomcat8080.tar |

【sept1】安装依赖环境

linux命令：

#cd /home/tools/docker1.7.1离线安装依赖包及自动安装脚本/ #进入目录

#rpm -ivh \*.rpm --force --nodeps # 执行目录下所以的rpm程序

【sept2】挂载cgroup

linux命令：

#vi /etc/fstab #编辑fatab文件

#在结尾添加

none /sys/fs/cgroup cgroup defaults 0 0

#保存重启

#reboot

【sept3】再次运行启动

#docker ps //看到container id等字样代表安装成功

//以不带网卡方式启动

#docker run -i -t --rm --net='none' ubuntu /bin/bash

6：registry私服搭建

6.1registry私服搭建

【sept1】载入文件registryV1.0.0H1.0.0.tar

#docker load --input /home/tools/loadMenu/registryV1.0.0H1.0.0.tar

#载入成功后，查看

#docker images

【sept2】启动registry私服容器服务

#docker run -d --net=host -v /etc/localtime:/etc/localtime:ro -v /home/tools/date/registry/:/var/lib/registry/ --name registry2 --restart=always registry:V1.0.0H1.0.0

私仓查询：

#curl <http://192.168.3.176:5000/v2/_catalog>

查询镜像的tag

# curl 192.168.3.176:5000/v2/sujun/tomcat/tags/list

{"name":"sujun/tomcat","tags":["v08302018"]}

【sept3】生成文件registry5000.tar

# docker save -o /home/tools/saveMenu/registry5000.tar registry:latest

【sept4】客户端docker上传配置

#修改客户端的属性：

# vi /etc/sysconfig/docker

other\_args=" --insecure-registry 192.168.3.205:5000"//添加配置ip：端口

#service docker stop //停止

#service docker start //启动

#ps -ef|grep docker //查看有192.168.3.205:5000

7：tomcat镜像服务启动

7.1tomcat镜像服务搭建

【sept1】载入文件tomcat8080.tar

#save load --input /home/tools/loadMenu/tomcat8080.tar

#载入成功后，查看

#docker images

【sept2】启动tomcat容器服务

#默认bridge模式;带network namespace和ip、端口

#docker run -d -p 8080:8080 -v /home/tools/data/tomcat/webapps:/usr/local/tomcat/webapps -v /home/tools/data/tomcat/logs/:/usr/local/tomcat/logs/ --name jwzh\_tomcat-rm tomcat8080

#以host方式启动；以宿主机的ip和端口为准。容器不带ip和端口

#docker run --net=host -d -v /home/tools/data/tomcat/webapps:/usr/local/tomcat/webapps -v /home/tools/data/tomcat/logs/:/usr/local/tomcat/logs/ --name jwzh\_tomcat-rm tomcat8080

【sept3】生成文件tomcat8080.tar

# docker save -o /home/tools/saveMenu/tomcat8080.tar tomcat8080:latest

8：mysql镜像服务启动

8.1mysql镜像服务搭建

【sept1】载入文件mysql-jwzh.tar

#save load --input /home/tools/loadMenu/mysql-jwzh.tar

#载入成功后，查看

#docker images

【sept2】启动mysql容器服务 --ulimit nofile=65536:65536 解决最大连接为214问题

#docker run -d -p 3306:3306 -v /home/tools/data/mysql/:/var/lib/mysql/ -e MYSQL\_ROOT\_PASSWORD=123456 --ulimit nofile=65536:65536 --name mysql-jwzh-rm mysql-jwzh:latest

【sept3】生成文件mysql3306.tar

# docker save -o /home/tools/saveMenu/mysql3306.tar mysql3306:latest

【sept4】登录并创建user用户

# 创建用户密码

# CREATE USER 'user'@'localhost' IDENTIFIED BY '123456';

# 授权

# GRANT ALL PRIVILEGES ON \*.\* TO 'user'@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;

#刷新权限

#flush privileges;

完成

9：redis镜像服务启动

9.1redis镜像服务搭建

【sept1】载入文件redis3.0.tar

# docker load --input /home/tools/tar包/redis3.0.tar

#载入成功后，查看

#docker images

【sept2】启动redis容器服务

#docker run -p 6379:6379 -v /home/tools/data/redis:/data -d --name redis-rm redis:3.0 redis-server --appendonly yes

【sept3】生成文件redis3.0.tar

# docker save -o /home/tools/saveMenu/redis3.0.tar redis:3.0

10：mongo镜像服务启动

10.1mongo镜像服务搭建

【sept1】载入文件mongo3.0.tar

# docker load --input /home/tools/tar包/mongo3.0.tar

#载入成功后，查看

#docker images

【sept2】启动mongo容器服务

#docker run -p 27017:27017 -v /home/tools/data/mongo/db:/data/db -d --name mongo-rm mongo:3.0.6

【sept3】生成文件mongo3.0.tar

# docker save -o /home/tools/saveMenu/mongo3.0.tar mongo:3.0.6

【sept4】进入容器创建用户密码和数据库

# docker exec -it 镜像id /bin/bash （进入容器）

# mongo （进入mongodb）

下面是安装完之后，正确的执行步骤，创建管理账户用户和普通用户的步骤，

创建管理账户，然后退出。

use admin  
 db.createUser(  
 {  
 user: "admin",  
 pwd: "password",  
 roles: [ { role: "root", db: "admin" } ]  
 }  
 );

exit; （退出容器）

use jwzh

exit;

11：rabbitmq镜像服务启动

11.1rabbitmq镜像服务搭建

【sept1】载入文件rabbitmq-3.6.1-management.tar

# docker load --input /home/tools/tar包/rabbitmq-3.6.1-management.tar

#载入成功后，查看

#docker images

【sept2】启动rabbitmq容器服务

#docker run -d --name myrabbitmq --restart=always -p 5672:5672 -p 15672:15672 rabbitmq:3.6.1-management

【sept3】生成文件rabbitmq-3.6.1-management.tar

# docker save -o /home/tools/saveMenu/rabbitmq-3.6.1-management.tar rabbitmq:3.6.1-management

【sept4】默认帐号密码都为:guest

12：nginx镜像服务启动

12.1nginx镜像服务搭建

【sept1】载入文件nginx1.10.tar

# docker load --input /home/tools/tar包/nginx1.10.tar

#载入成功后，查看

#docker images

【sept2】启动nginx容器服务

#docker run -p 80:80 --name mynginx --restart=always -v /home/tools/data/nginx-gridfs/www:/www -v /home/tools/data/nginx-gridfs/logs:/wwwlogs -d nginx:1.10.3

【sept3】生成文件nginx1.10.tar

# docker save -o /home/tools/saveMenu/nginx1.10.tar nginx:1.10.3

13：centos-nginx-gridfs镜像服务启动

13.1centos-nginx-gridfs镜像服务搭建

【sept1】载入文件centos-nginx-gridfs.tar

# docker load --input /home/tools/tar包/centos-nginx-gridfs.tar

#载入成功后，查看

#docker images

【sept2】启动centos-nginx-gridfs容器服务

#docker run -itd -p 80:80 --name centos-nginx-gridfs-rm --restart=always centos-nginx-gridfs /home/auto\_service.sh

【sept3】生成文件centos-nginx-gridfs.tar

#docker save -o /home/tools/tar包/centos-nginx-gridfs.tar centos-nginx-gridfs

【sept4】步骤

#载入centos6.9.tar

#docker load --input /home/tools/centos6.9.tar

#启动容器

#docker run -it -name centos-nginx centos:6.9

#查容器 信息

# docker ps

#查容器ID对应的32位ID

# docker inspect -f '{{.Id}}' 15cf7b611117

#拷贝文件到mycentos容器中

# cp /home/tools/nginx-gridfs.tar.gz /var/lib/docker/devicemapper/mnt/15cf7b6111173395f583d657458982ca53aedf54516fae6e8cfc52ea84e0e77a/rootfs/mnt

# cp /home/tools/nginx-gridfs-rpm.tar.gz /var/lib/docker/devicemapper/mnt/15cf7b6111173395f583d657458982ca53aedf54516fae6e8cfc52ea84e0e77a/rootfs/mnt

#进入容器

# docker exec -it centos-nginx /bin/bash

拷贝到usr的local下

# cp /mnt/nginx-gridfs.tar.gz /usr/local/

# cp /mnt/nginx-gridfs-rpm.tar.gz /usr/local/

#解压

#cd /usr/local/

#tar -zxvf nginx-gridfs.tar.gz

#tar -zxvf nginx-gridfs-rpm.tar.gz

#安装rpm

#rpm -ivh \*.rpm --force --nodeps

#授权

#cd /usr/local/nginx-gridfs/sbin/

#chmod 777 nginx

#nginx 启动测试

#/usr/local/nginx-gridfs/sbin/nginx -c /usr/local/nginx-gridfs/conf/nginx.conf

#设置自动重启

Vi /home/auto\_service.sh

#添加一下信息

#!/bin/sh

/usr/local/nginx-gridfs/sbin/nginx -c /usr/local/nginx-gridfs/conf/nginx.conf

/bin/bash

#保存授权：wq

Chmod 777 /home/auto\_service.sh

#根据容器打镜像

docker commit ce88f2ba643c centos-nginx-grifds

#运行容器带开机启动脚本

#docker run -itd -p 80:80 --name centos-nginx-gridfs2 centos-nginx-gridfs-v2 /home/auto\_service.sh

#启动脚本：

/home/auto\_service.sh

14：harbor搭建docker私有仓库

14.1harbor服务搭建

【sept1】安装Docker-compose

# 内网服务器环境(无法连接互联网)中安装docker-compose

在内网环境中，执行docker-compose安装程序并适时终止；即执行   
 $ curl -L <https://github.com/docker/compose/releases/download/1.1.0/doc> ker-compose-uname -s-uname -m` > /usr/local/bin/docker-compose   
 将外网环境中安装docker-compose后的安装文件/usr/local/bin/docker-compos e拷贝至内网环境中同路径下，并执行

$ chmod a+x /usr/local/bin/docker-compose   
 后执行docker-compose,执行结果如下，则安装成功。

$ docker-compose

【sept2】启动nginx容器服务

#docker run -p 80:80 --name mynginx --restart=always -v /home/tools/data/nginx-gridfs/www:/www -v /home/tools/data/nginx-gridfs/logs:/wwwlogs -d nginx:1.10.3

【sept3】生成文件nginx1.10.tar

# docker save -o /home/tools/saveMenu/nginx1.10.tar nginx:1.10.3