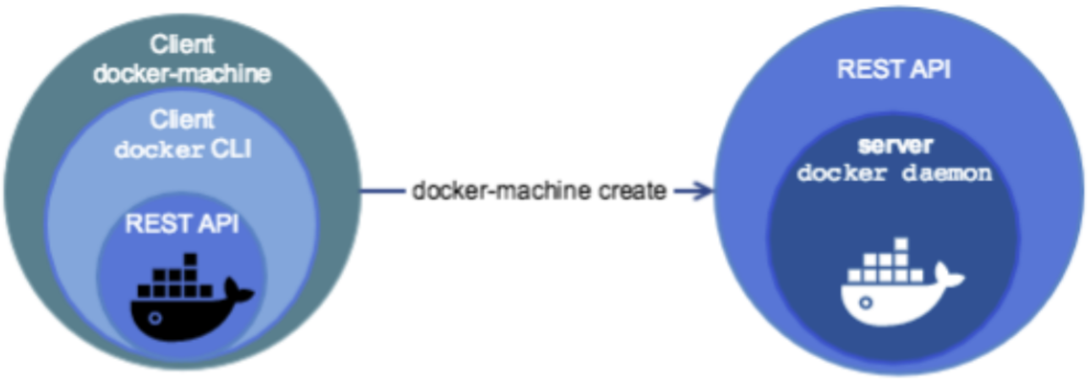
Docker Machine

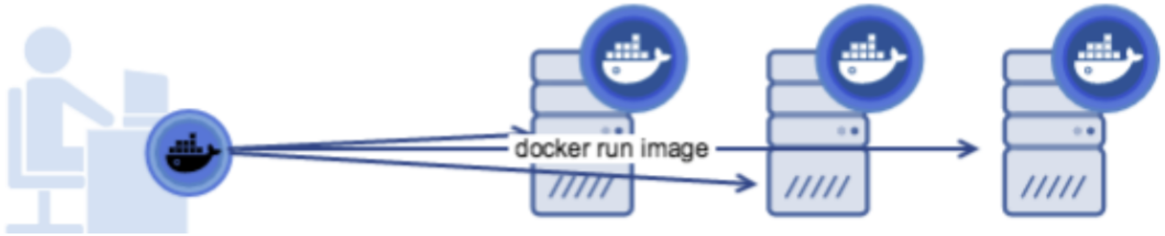
https://zhuanlan.zhihu.com/p/358666246

Docker Machine可用于配置和管理您的Dockerized主机（主机上有Docker Engine）。

通常在本地系统上安装Docker Machine。Docker Machine有自己的命令行客户端docker-machine和Docker Engine客户端docker。您可以使用machine在一个或多个虚拟系统上安装Docker Engine。这些虚拟系统可以是本地的（如在Mac或Windows上使用machine在VirtualBox中安装和运行Docker Engine）或远程的（如在云提供商上使用machine安装和运行Docker Engine）。

Docker Machine





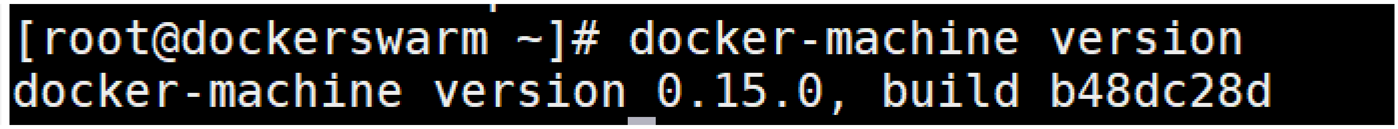
Linux安装(Centos7)

在终端执行下列命令：

```
$ base=http://mirrors.aliyun.com/docker-toolbox/linux/machine/0.15.0 &&
  curl -L $base/docker-machine-Linux-x86_64 >/tmp/docker-machine &&
  sudo mv /tmp/docker-machine /usr/local/bin/docker-machine &&
  chmod +x /usr/local/bin/docker-machine
```

检查是否安装成功

```
docker-machine version
```



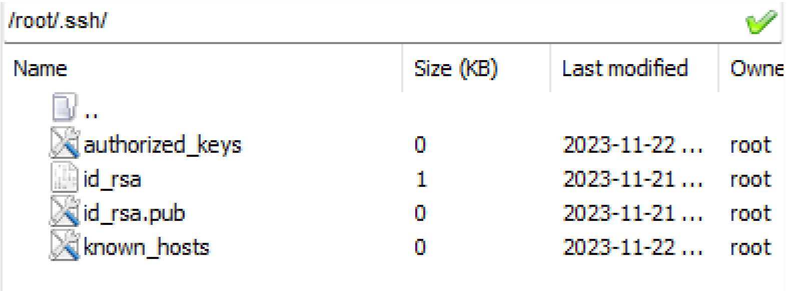
```
[root@dockerswarm ~]# docker-machine version
docker-machine version 0.15.0, build b48dc28d
```

SSH

生成宿主机ssh钥匙🔑

```
# 当前主机地址为 root@172.22.70.12
ssh-keygen # 随后直接无脑回车 （以下显示结果）
```

```
[root@dockerswarm ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:YCm7jwq5oDEq7XHDLMUqNNO6bAL+wfTgpJHeB+6s77c root@dockerswarm
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|       .         |
|    . +          |
|  o. + .         |
| * *+   S        |
|+.&=+.           |
|X=+**o           |
|BO=+o=           |
|B=BBoEo          |
+----[SHA256]-----+
id_rsa.pub : 公钥
id_rsa :  私钥
```

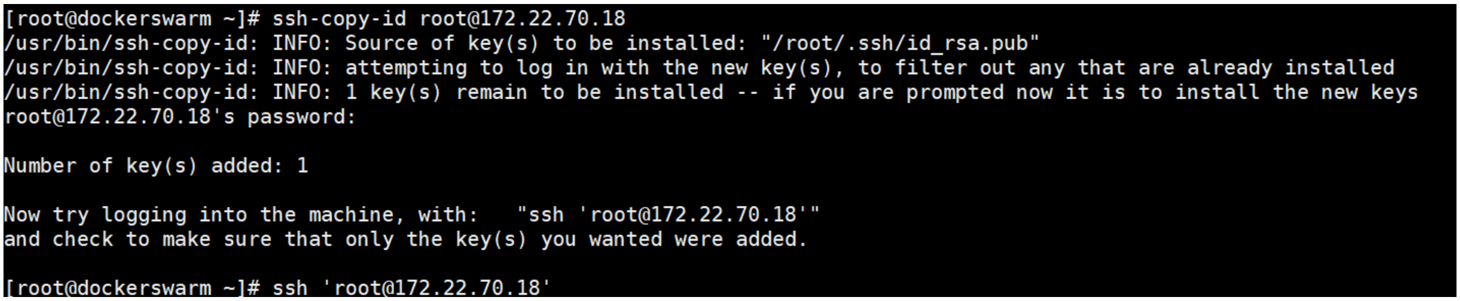


/root/.ssh/

Name	Size (KB)	Last modified	Owne
..			
authorized_keys	0	2023-11-22 ...	root
id_rsa	1	2023-11-21 ...	root
id_rsa.pub	0	2023-11-21 ...	root
known_hosts	0	2023-11-22 ...	root

SSH钥匙复制到目标主机

```
ssh-copy-id <用户名>@<目标主机ip地址>
# 可通过ip addr 查看 ens192
# 示例：当前主机地址172.22.70.12复制ssh钥匙到目标主机172.22.70.18
ssh-copy-id root@172.22.70.18
```



```
[root@dockerswarm ~]# ssh-copy-id root@172.22.70.18
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.22.70.18's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@172.22.70.18'"
and check to make sure that only the key(s) you wanted were added.

[root@dockerswarm ~]# ssh 'root@172.22.70.18'
```

SSH登录目标主机

复制ssh key到目标主机成功后, 可通过ssh命令直接登入目标主机（172.22.70.18）：

```
ssh 'root@172.22.70.18'
```

Docker Machine创建

直接创建

如果本机没有虚拟机，直接使用create方法创建docker machine

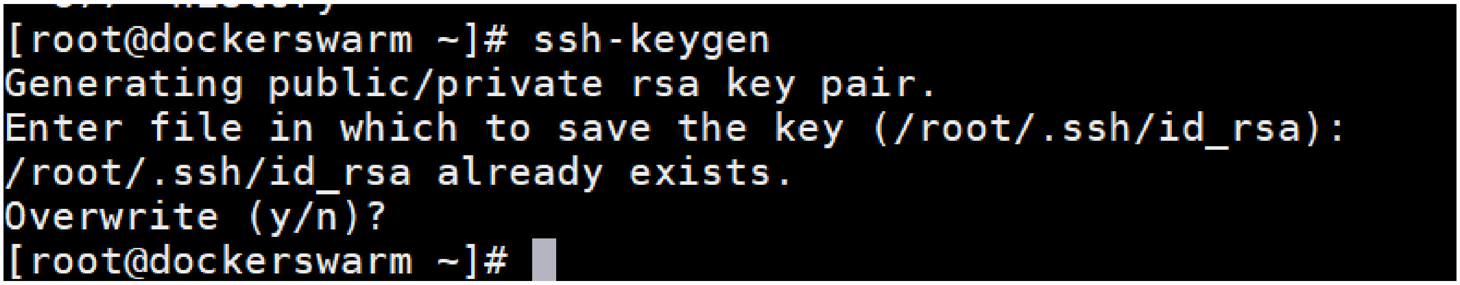
https://zhuanlan.zhihu.com/p/35102874

```
docker-machine create default --virtualbox-no-vtx-check
# 禁止vtx检查，避免出现 Error with pre-create check: "This computer doesn't have VT-X/AMD-v enabl
```

已存在添加

如果本机已经有虚拟机之后，直接使用create方法将目标docker添加到docker machine
https://blog.csdn.net/m0_46243410/article/details/107690124

```
# ssh key 生成方式：
ssh-keygen
# /root/.ssh/id_rsa
# 将ssh key copy 到目标主机
ssh-copy-id root@172.22.70.18
```



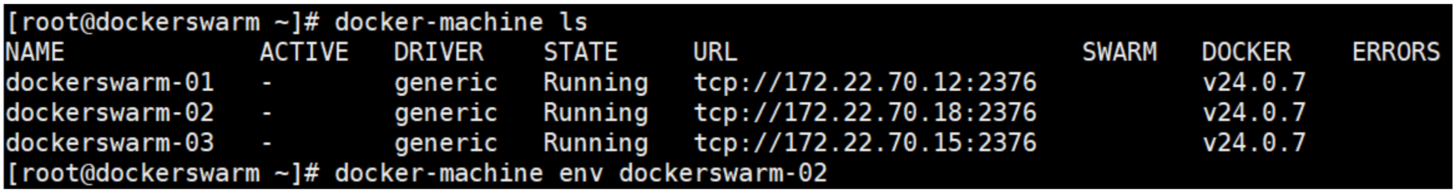
```
[root@dockerswarm ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)?
[root@dockerswarm ~]#
```

执行下面命令，将目标docker主机添加到docker machine宿主机

```
docker-machine create --driver generic --generic-ip-address=172.22.70.18 --generic-ssh-key ~/.s
# 注意目标主机一定要开启docker才行连接
```

在docker machine宿主机，可以看到添加的目标docker主机

```
docker-machine ls
```



```
[root@dockerswarm ~]# docker-machine ls
NAME              ACTIVE   DRIVER    STATE     URL                        SWARM   DOCKER    ERRORS
dockerswarm-01    -        generic   Running   tcp://172.22.70.12:2376            v24.0.7
dockerswarm-02    -        generic   Running   tcp://172.22.70.18:2376            v24.0.7
dockerswarm-03    -        generic   Running   tcp://172.22.70.15:2376            v24.0.7
[root@dockerswarm ~]# docker-machine env dockerswarm-02
```

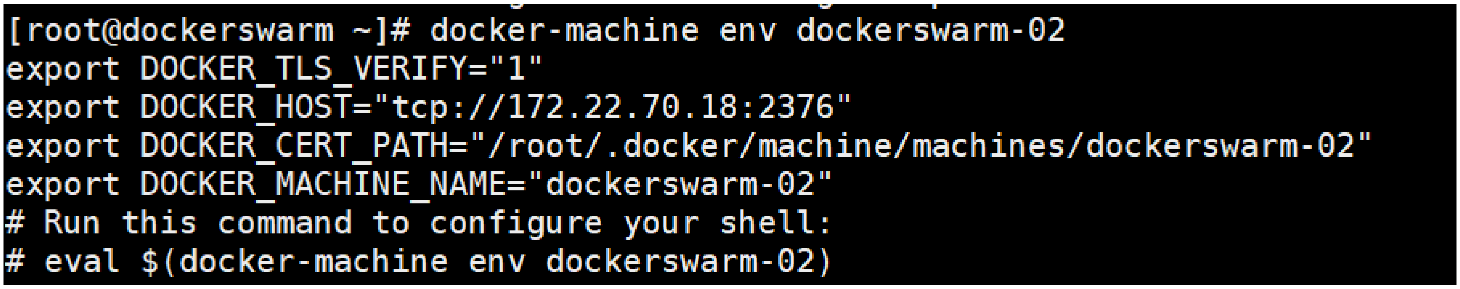
管理操作目标Machine

进入目标机Docker环境

```
docker-machine env dockerswarm-02
```

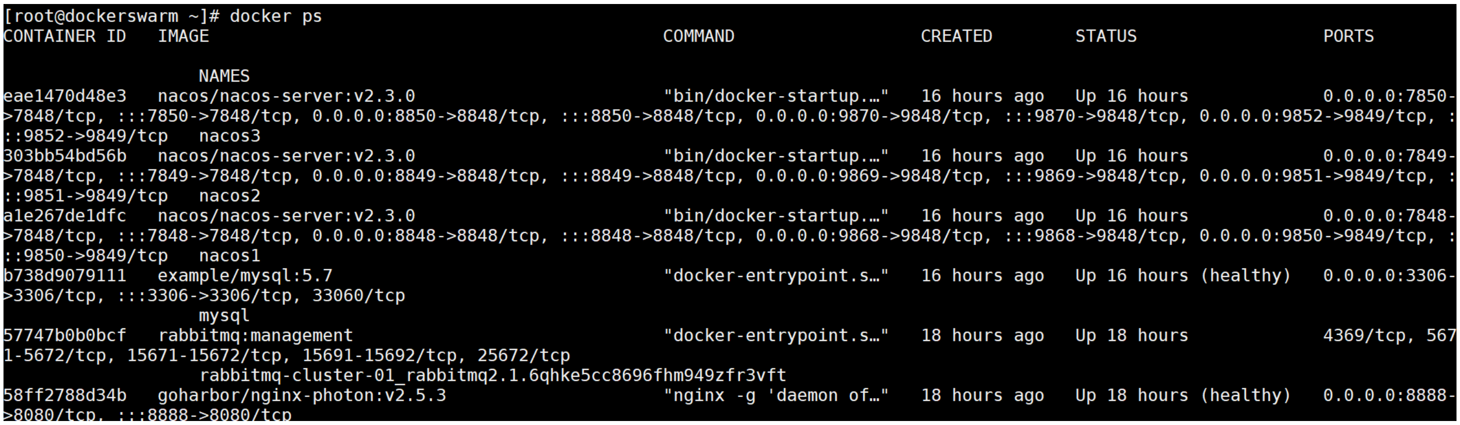
提示执行以下命令配置环境

```
eval $(docker-machine env dockerswarm-02)
```



```
[root@dockerswarm ~]# docker-machine env dockerswarm-02
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://172.22.70.18:2376"
export DOCKER_CERT_PATH="/root/.docker/machine/machines/dockerswarm-02"
export DOCKER_MACHINE_NAME="dockerswarm-02"
# Run this command to configure your shell:
# eval $(docker-machine env dockerswarm-02)
```

执行完成后使用docker命令，即可操作目标主机的docker



```
[root@dockerswarm ~]# docker ps
CONTAINER ID   IMAGE                                            COMMAND                  CREATED        STATUS                   PORTS

                   NAMES
eae1470d48e3   nacos/nacos-server:v2.3.0                        "bin/docker-startup.…"   16 hours ago   Up 16 hours              0.0.0.0:7850-
>7848/tcp, :::7850->7848/tcp, 0.0.0.0:8850->8848/tcp, :::8850->8848/tcp, 0.0.0.0:9870->9848/tcp, :::9870->9848/tcp, 0.0.0.0:9852->9849/tcp, :
::9852->9849/tcp   nacos3
303bb54bd56b   nacos/nacos-server:v2.3.0                        "bin/docker-startup.…"   16 hours ago   Up 16 hours              0.0.0.0:7849-
>7848/tcp, :::7849->7848/tcp, 0.0.0.0:8849->8848/tcp, :::8849->8848/tcp, 0.0.0.0:9869->9848/tcp, :::9869->9848/tcp, 0.0.0.0:9851->9849/tcp, :
::9851->9849/tcp   nacos2
a1e267de1dfc   nacos/nacos-server:v2.3.0                        "bin/docker-startup.…"   16 hours ago   Up 16 hours              0.0.0.0:7848-
>7848/tcp, :::7848->7848/tcp, 0.0.0.0:8848->8848/tcp, :::8848->8848/tcp, 0.0.0.0:9868->9848/tcp, :::9868->9848/tcp, 0.0.0.0:9850->9849/tcp, :
::9850->9849/tcp   nacos1
b738d9079111   example/mysql:5.7                                "docker-entrypoint.s…"   16 hours ago   Up 16 hours (healthy)   0.0.0.0:3306-
>3306/tcp, :::3306->3306/tcp, 33060/tcp
                   mysql
57747b0b0bcf   rabbitmq:management                              "docker-entrypoint.s…"   18 hours ago   Up 18 hours              4369/tcp, 567
1-5672/tcp, 15671-15672/tcp, 15691-15692/tcp, 25672/tcp
                   rabbitmq-cluster-01_rabbitmq2.1.6qhke5cc8696fhm949zfr3vft
58ff2788d34b   goharbor/nginx-photon:v2.5.3                     "nginx -g 'daemon of…"   18 hours ago   Up 18 hours (healthy)   0.0.0.0:8888-
>8080/tcp, :::8888->8080/tcp
```