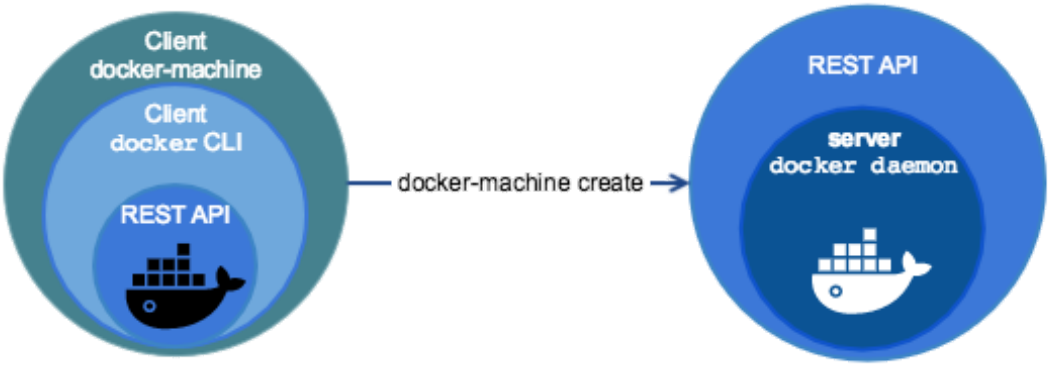
Docker Compose

```
# 安装
sudo curl -L https://github.com/docker/compose/releases/download/1.18.0/docker-
# 创建工作目录
# 编写docker-compose.yaml,创建wordpress的案例。
docker-compose up -d

version: '3'
services:
   db:
     image: mysql:5.7
     volumes:
       - db_data:/var/lib/mysql
     restart: always
     environment:
       MYSQL_ROOT_PASSWORD: somewordpress
       MYSQL_DATABASE: wordpress
       MYSQL_USER: wordpress
       MYSQL_PASSWORD: wordpress

   wordpress:
     depends_on:
       - db
     image: wordpress:latest
     ports:
       - "8000:80"
     restart: always
     environment:
       WORDPRESS_DB_HOST: db:3306
       WORDPRESS_DB_USER: wordpress
       WORDPRESS_DB_PASSWORD: wordpress
volumes:
    db_data:

```

Docker Machine

帮助用户在不同的云平台上创建和管理虚拟机，并且在虚拟机中安装Docker引擎，通过Macheine可以管理Docker宿主机（https://docs.docker.com/machine/overview/）。




```
curl -L https://github.com/docker/machine/releases/download/v0.13.0/docker-mach

# 关于云平台，可以参考：https://docs.docker.com/machine/drivers/
# 配置以下环境变量后就可对接Openstack。
# 对接部门1.0版本的过程中发现了一些问题：

# 1.部门的所有aggregate均没有配置Zone，且aggregate中包含KVM和VMware两种平台。导致do
# 2.从web前台删除虚机时，会跟着删除flavor，导致每次创建虚机都要创建配额。


export OS_AVAILABILITY_ZONE=nova
export OS_DOMAIN_NAME=default
export OS_PROJECT_DOMAIN_NAME=default
export OS_USER_DOMAIN_NAME=default
export OS_PROJECT_NAME=LinQ
export OS_TENANT_NAME=LinQ
export OS_USERNAME=linq
export OS_PASSWORD=Ruijie@123
```

```
export OS_AUTH_URL=http://172.24.9.198:35357/v3
export OS_INTERFACE=internal
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_ID=6fa06b5e-5aae-433e-9753-2577ebd268f1
export OS_FLAVOR_ID=33fe0d69-9af9-470f-b3c3-f71002608eb6
export OS_NETWORK_ID=66245dd4-0e14-4f59-bb10-9986b9910513

```

Docker Swarm

在Docker 1.12及更高版本中，swarm模式与Docker引擎集成在一起。但是需要注意的是相关的API没有集成到docker API server或者CLI。

Swarm的主要目标是：将主机池变成一个单一的虚拟主机，并且swarm的API兼容原生docker API，使原有的应用可以无缝的迁移到swarm集群。

```

# 安装etcd、etcd-browser
Docker17 :172.24.10.3
# 安装swarm manage
vm-m :172.24.10.64
# 安装swarm agent
vm-1: 172.24.10.7 vm-2: 172.24.10.56

# Docker17 执行etcd安装启动命令
export HostIP=172.24.10.3
docker run -d -v /usr/share/ca-certificates/:/etc/ssl/certs -p 4001:4001 -p 238
--restart=always \
--name etcd quay.io/coreos/etcd \
/usr/local/bin/etcd \
-name etcd0 \
-advertise-client-urls http://${HostIP}:2379,http://${HostIP}:4001 \
-listen-client-urls http://0.0.0.0:2379,http://0.0.0.0:4001 \
-initial-advertise-peer-urls http://${HostIP}:2380 \
-listen-peer-urls http://0.0.0.0:2380 \
-initial-cluster-token etcd-cluster-1 \
-initial-cluster etcd0=http://${HostIP}:2380 \
-initial-cluster-state new
```

```
22
24 docker run --rm --name etcd-browser -d -p 38000:8000 --env ETCD_HOST=172.24.10.
25
26 # vm-m 启动swarm manage
27 docker run -dit --name vm-m -p 0.0.0.0:3376:3376 --restart=always -v /etc/docke
28 swarm manage -H tcp://0.0.0.0:3376  etcd://172.24.10.3:2379/swarm  --strategy s
29 # 添加vm-1、vm-2到集群
30 docker run -d --name vm-1 --restart=always swarm join --advertise=172.24.10.7:2
31 docker run -d --name vm-2 --restart=always swarm join --advertise=172.24.10.56:
32
33 # 在Docker17 验证集群信息
34 #注意：Docker17上的3376端口被暴露给swarm容器，所以当指定了DOCKER_HOST到这个端口之后，
35 export DOCKER_HOST=172.24.10.64:3376
36 docker info # 此时应该能看到节点信息
37
38 # 注意上面的这种方法，没有开启TLS，因此如果HOST开启了TLS会使manger拿不到agent的信息。
```

```
1  # 创建master
2  docker-machine create -d openstack --swarm --swarm-master --swarm-discovery etc
3  # 添加agent
4  docker-machine create -d openstack --swarm --swarm-discovery etcd://172.24.10.3
5  # 如果已经创建了机器想加入到集群中，那么可以编辑machine的config.json，修改相应配置项。
6  {
7      "ConfigVersion": 3,
8      "Driver": {
9          "SwarmMaster": true,
10         "SwarmHost": "tcp://0.0.0.0:3376",
11         "SwarmDiscovery": "etcd://172.24.10.3:2379",
12     },
13     "DriverName": "openstack",
14     "HostOptions": {
15         /* 关于节点信息的配置 */
16         },
17         "SwarmOptions": {
18             "IsSwarm": true,
19             "Address": "",
```

```
            "Discovery": "etcd://172.24.10.3:2379",
            "Agent": true,
            "Master": true,
            "Host": "tcp://0.0.0.0:3376",
            "Image": "swarm:latest",
            "Strategy": "spread",
            "Heartbeat": 0,
            "Overcommit": 0,
            "ArbitraryFlags": [],
            "ArbitraryJoinFlags": [],
            "Env": null,
            "IsExperimental": false
        },
        "AuthOptions": {
        /* 关于Cert证书的配置 */
        }
    },
    "Name": "vm-m-test"
}

# 执行provision命令，重新配置
docker-machine provision <machine-name>
# 获取集群环境变量
docker-machine env --swarm vm-m
```

```
# swarm服务发现。swarm支持多种后端，用于服务发现：https://docs.docker.com/swarm/dis
1. 键值数据库，libkv（https://github.com/docker/libkv）这个项目建立一个抽象层，使sw
2.docker hub，使用swarm create命令能够将发现服务托管到docker hub。
```