Docker and Vagrant

宁哥

http://www.lining0806.com/

介绍

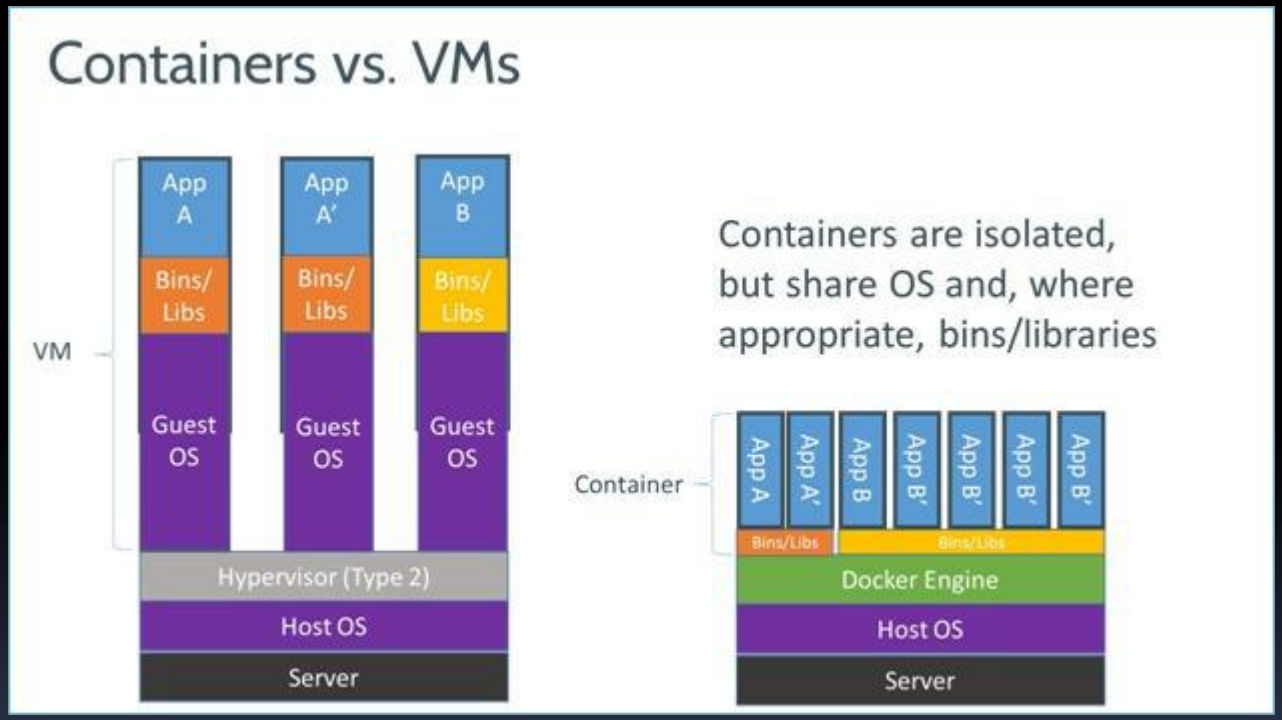
- Docker

  运行环境部署工具
  基于linux container
  Docker - Build, Ship, and Run Any App, Anywhere.

- Vagrant

  开发环境部署工具
  基于virtualbox
  Vagrant - Development environments made easy.

Docker



- 容器化

  容器内外进程隔离，容器之内进程可见

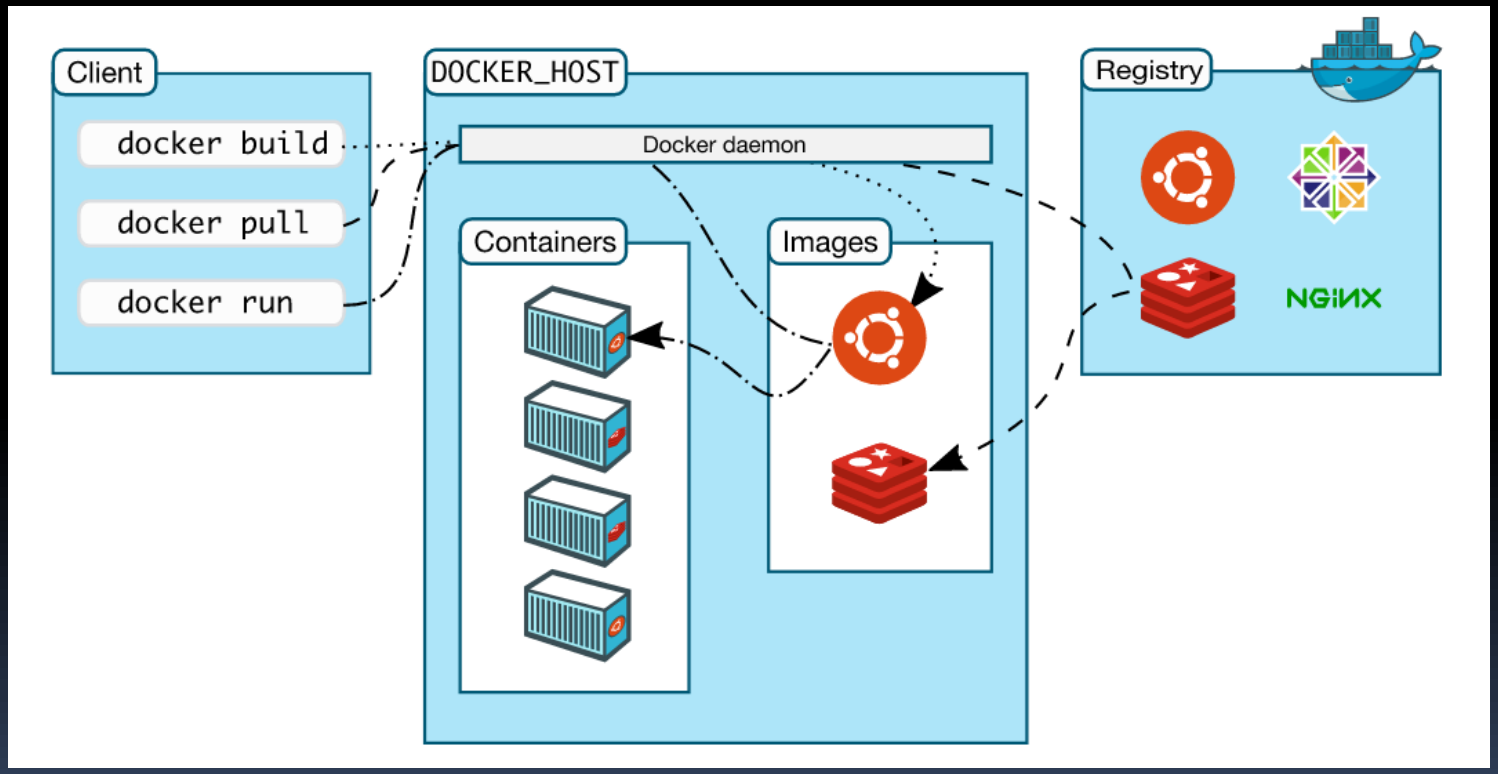
- Docker优点

  启动速度快

  资源利用率高，普通桌面机可以运行很多台容器

  性能开销小，引用共用宿主机操作系统

Docker


Client
docker build
docker pull
docker run
DOCKER_HOST
Docker daemon
Containers
Images
Registry
NGINX

Docker

```
# root @ ubuntu64 in ~ [16:34:24]
$ docker -h
Flag shorthand -h has been deprecated, please use --help

Usage:  docker COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client con
  -D, --debug              Enable debug mode
      --help               Print usage
  -H, --host list          Daemon socket(s) to co
  -l, --log-level string   Set the logging level
      --tls                Use TLS; implied by --
      --tlscacert string   Trust certs signed onl
      --tlscert string     Path to TLS certificat
      --tlskey string      Path to TLS key file (
      --tlsverify          Use TLS and verify the
  -v, --version            Print version informat

Commands:
  attach      Attach to a running container
  build       Build an image from a Dockerfile
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes on a container's filesystem
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  images      List images
  import      Import the contents from a tarball to create a filesystem image
  info        Display system-wide information
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  login       Log in to a Docker registry
  logout      Log out from a Docker registry
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  ps          List containers
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  run         Run a command in a new container
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
```

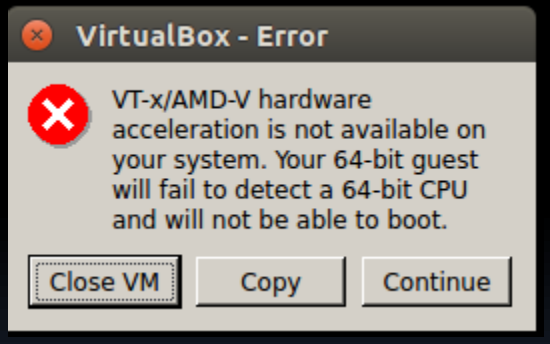
Docker操作演示

- Docker应用

使用Docker容器开发、测试、部署服务
创建隔离的运行环境
搭建测试环境

Vagrant

Host：Win7 64
Provider：Virtualbox
Box：http://www.vagrantbox.es/



- 硬件加速：

进入BIOS开启CPU的硬件虚拟化功能，虚拟化技术的选项名称大概含有“virtualization technology”的字眼。
现在的CPU几乎都支持硬件虚拟化功能，英特尔称之为VT-x技术，AMD称之为AMD-V技术。

Vagrant

```
C:\Users\FIRELING\Desktop\vagrant_test\base>vagrant --version
Vagrant 1.9.1

C:\Users\FIRELING\Desktop\vagrant_test\base>vagrant box add base trusty-server-cloudimg-amd64-vagrant-disk1.box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'base' (v0) for provider:
    box: Unpacking necessary files from: file://C:/Users/FIRELING/Desktop/vagrant_test/base/trusty-server-cloudimg-amd64-vagrant-disk1.box
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
    box:
x box.ovf
x Vagrantfile
x box-disk1.vmdk==> box: Successfully added box 'base' (v0) for 'virtualbox'!

C:\Users\FIRELING\Desktop\vagrant_test\base>vagrant init base
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
```

- 配置文件Vagrantfile：

Vagrant可以通过编写一个Vagrantfile来控制虚拟机的启动、虚拟机网络环境的配置、虚拟机与主机间的文件共享，以及启动后自动执行一些配置脚本，比如自动执行一个shell script来安装一些必备的开发工具，如Mysql。

这意味着，当你需要在多台机器间同步开发进度时，只需要同步Vagrantfile，就可以保证各台机器拥有一致的开发环境。

Vagrant

```
## 单机配置

config.vm.box = "base"
# config.vm.network "forwarded_port", guest: 80, host: 8080
config.vm.network "private_network", ip: "192.168.33.10"
# config.vm.network "public_network"
# config.vm.synced_folder "../data", "/vagrant_data"
config.vm.provider "virtualbox" do |vb|
    vb.customize ["modifyvm", :id, "--name", "ubuntu", "--memory", "1024"]
end
```

```
## 集群配置

config.vm.define :web do |web|
    web.vm.provider "virtualbox" do |v|
        v.customize ["modifyvm", :id, "--name", "web", "--memory", "1024"]
    end
    web.vm.box = "base"
    web.vm.hostname = "web"
    web.vm.network :private_network, ip: "192.168.33.10"
end

config.vm.define :redis do |redis|
    redis.vm.provider "virtualbox" do |v|
        v.customize ["modifyvm", :id, "--name", "redis", "--memory", "1024"]
    end
    redis.vm.box = "base"
    redis.vm.hostname = "redis"
    redis.vm.network :private_network, ip: "192.168.33.11"
end
```

Vagrant

1 vagrant1

```
vagrant@web:~$ ls
vagrant@web:~$ cd /
vagrant@web:/$ ls
bin   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  vagrant  vmlinuz
boot  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  var
vagrant@web:/$ hostname
web
vagrant@web:/$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1b:9b:de
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1b:9bde/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:666 errors:0 dropped:0 overruns:0 frame:0
          TX packets:476 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:80235 (80.2 KB)  TX bytes:65326 (65.3 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:3a:e7:a3
          inet addr:192.168.33.11  Bcast:192.168.33.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe3a:e7a3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:125 errors:0 dropped:0 overruns:0 frame:0
          TX packets:118 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12797 (12.7 KB)  TX bytes:14983 (14.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1296 (1.2 KB)  TX bytes:1296 (1.2 KB)

vagrant@web:/$
```

1 vagrant2

```
vagrant@redis:~$ ls
vagrant@redis:~$ cd /
vagrant@redis:/$ ls
bin   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  vagrant  vmlinuz
boot  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  var
vagrant@redis:/$ hostname
redis
vagrant@redis:/$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1b:9b:de
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1b:9bde/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:753 errors:0 dropped:0 overruns:0 frame:0
          TX packets:551 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:85305 (85.3 KB)  TX bytes:69702 (69.7 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:bc:48:46
          inet addr:192.168.33.12  Bcast:192.168.33.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:febc:4846/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:202 errors:0 dropped:0 overruns:0 frame:0
          TX packets:179 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20057 (20.0 KB)  TX bytes:23373 (23.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1296 (1.2 KB)  TX bytes:1296 (1.2 KB)

vagrant@redis:/$
```

Vagrant操作演示

http://www.vagrantbox.es/

下载box

打个打包好的系统在Vagrant中称为box，它实际上是一个zip包，可以在 http://www.vagrantbox.es/ 上找到你想要的box，下载到本地，或者直接copy对应box的url，以备vagrant添加box的时候使用。

添加box

```
vagrant box add {title} {url}
```

其中，title表示虚拟机镜像的名字，默认为base，url可以为本地box路径，也可以为url地址，支持在线安装。

初始化虚拟机

```
vagrant init {title}
```

此时会在本地生成一个名为Vagrantfile的文件，里面包含了Vagrant的配置信息，可以对虚拟机进行相应的配置。

启动虚拟机

```
vagrant up
```

第一次启动花费时间比较长。

连接虚拟机

```
vagrant ssh
```

登录之后就可以进行一系列操作了。

谢谢！