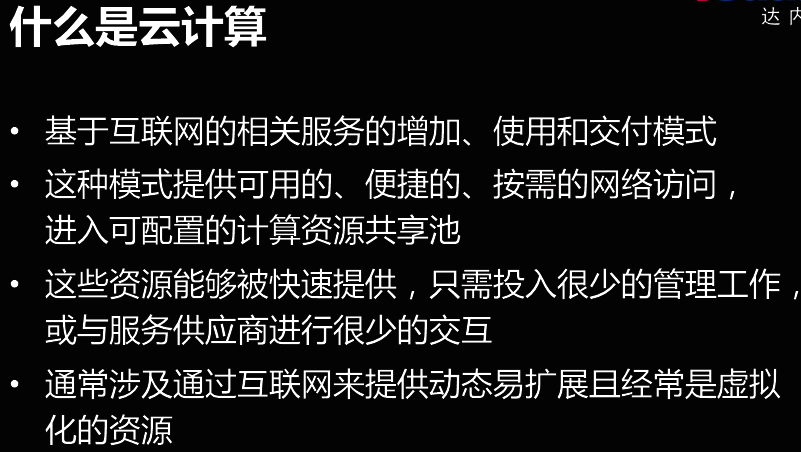
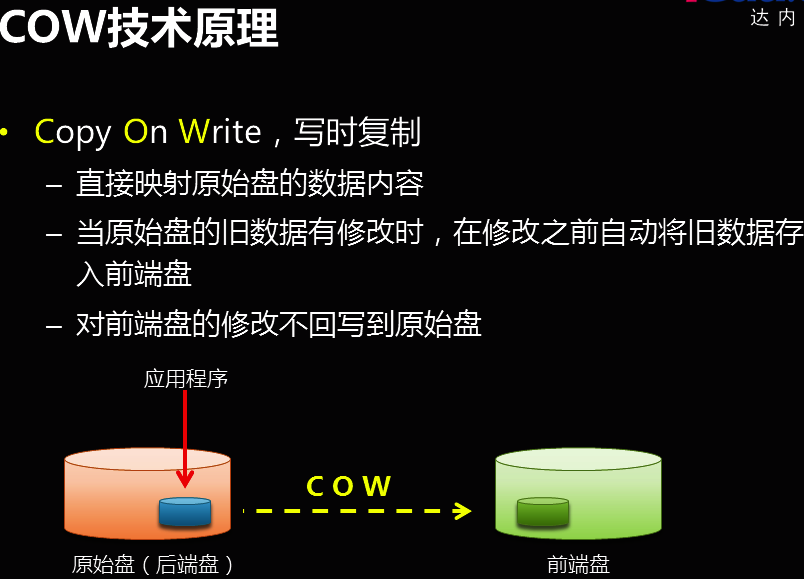
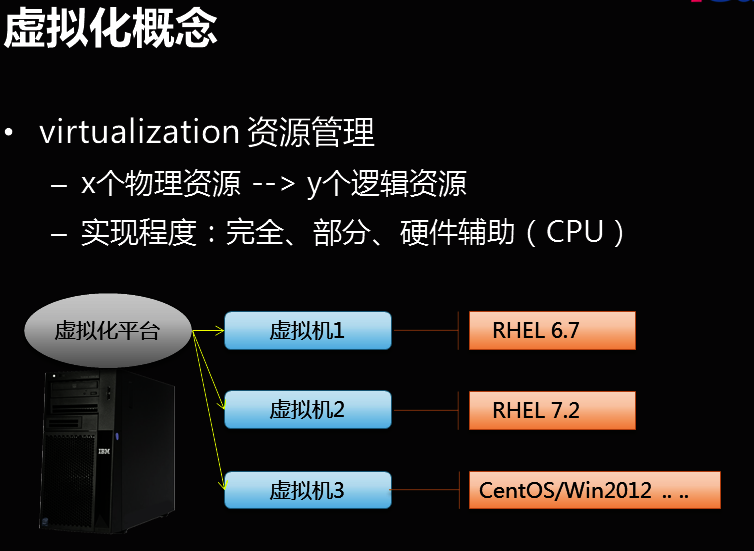
**Cloud复习**







使用KVM技术！！！

1.需要内和的支持 egrep '(vmx|svm)' /proc/cpuinfo

2.需要装4个包 qemu-kvm,libvirt-daemon,libvirt-client,lib-daemon-driver-qemu.

容器的核心技术组成？

Cfroups（Control Groups）资源管理

NameSpace进程隔离

SELINUX安全

Docker的优点：

相比于传统的虚拟化技术，容器更加简洁高效

容器使用的共享公共库和程序。

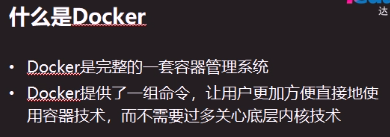
Docker的缺点

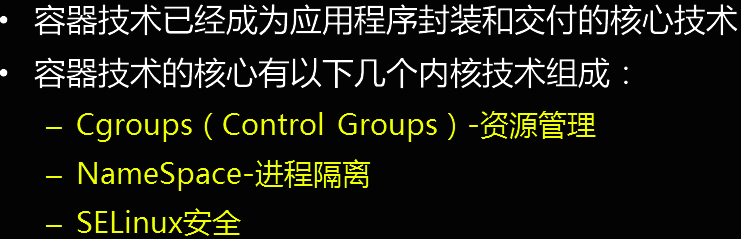
容器的个理性没有虚拟化强

共用底层的程序以及设备，安全性有先天缺陷

监控容器和容器派错是挑战

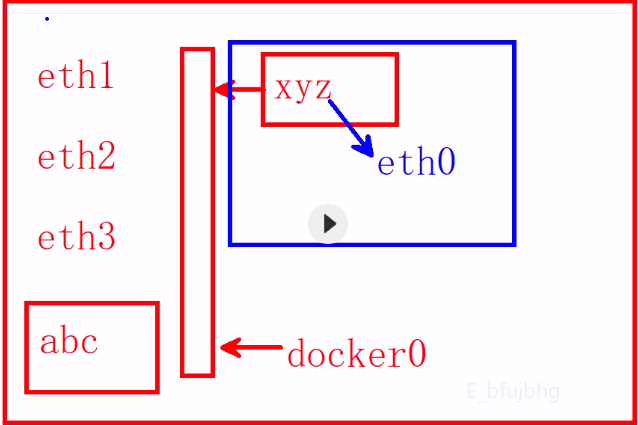
Selinux难以驾驭







Docker是一套容器管理系统，真正提供服务的是内核，和KVM的libvirt是同样的功能，提供与内核的交互。



**左边真实机，中间是虚拟出来的交换机，右边是容器。**

区分容器与真机：名称不一样，网络不一样，进程，挂载点，信号量，用户的认证

进程、信号量，用户的认证都是容器范围小。

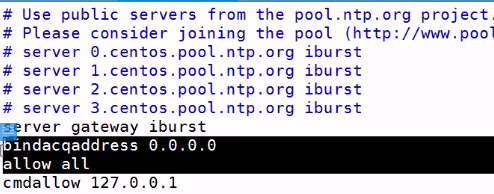
Kill其实是发送信号，

常用命令docker search pull（下载） push（上传）

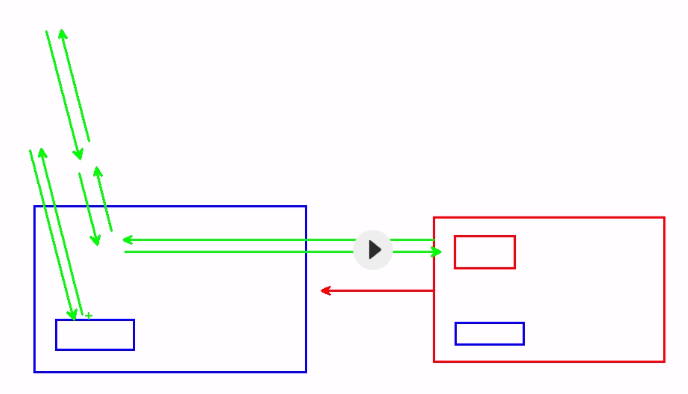
导入镜像 docker load < .\*.tar

导出镜像 docker save 镜像名 > .\*.tar

**时间服务器的配置**



本地虚拟化的时候，新创的虚拟机的前端盘几乎没有数据，只有当你用到的时候才会去后端盘调用，当修改过或者查看过之后就有数据了，就不会再去后端盘中取数据，他会有一个判断。



**之前作httpd容器的时候错了：**

1. 配置文件尽量不要用sed改，而是将文件复制到当前创建镜像的目录下。
2. www.server.com这一行，没有解析，改为localhost就好了。

**Docker启动ip是172.17.x.x**

**Tag是有一个起别名，打标记的作用。**

**自定义镜像**

@1.启动一个容易修修改该。退出

Docker commit 仓库ID name:label

@2.也可以使用dockerfile制作一个自定义的镜像，如nginx，httpd。

**自定义仓库**



真机开nfs共享！！



Nginx -T



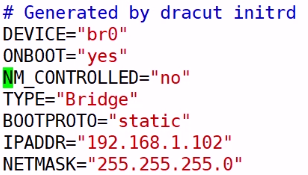
Docker2挂载



docker2新修改/var/webroot/index.html配置文件，docker1也会变。

**创建一个虚拟交换机（桥）**

首先要安装支持桥的包birdge-utils



NM\_controlled是取消nmcli控制。

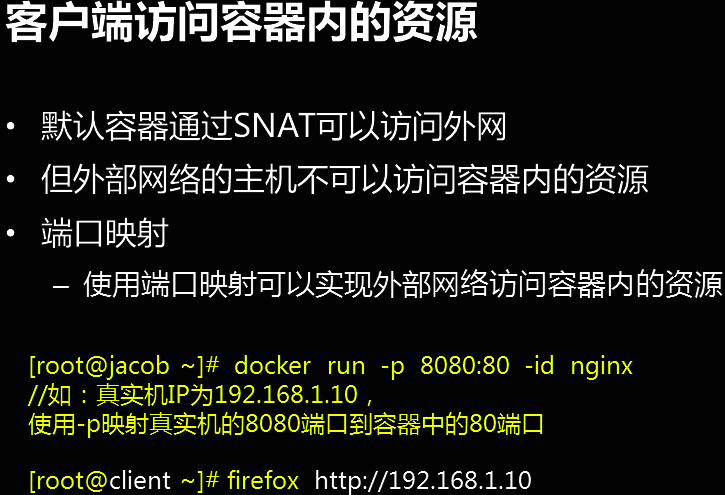
取消桥的时候ifconfig br0 down

然后删除配置文件，重起服务。

创建虚拟交换机：







**客户端要想访问容器内的资源，必须要有一个端口的映射。**



我真机分别起httpd和nginx，访问我的真机表头curl -i IP 分别得到的是不同的浏览器