## 海云平台容器化部署方案

一. 使用kubernetes平台管理容器

1.kubernetes介绍

Kubernetes是Google开源的容器集群管理系统，其提供应用部署、维护、 扩展机制等功能，利用Kubernetes能方便地管理跨机器运行容器化的应用，其主要功能如下：

（1）使用Docker对应用程序包装(package)、实例化(instantiate)、运行(run)。

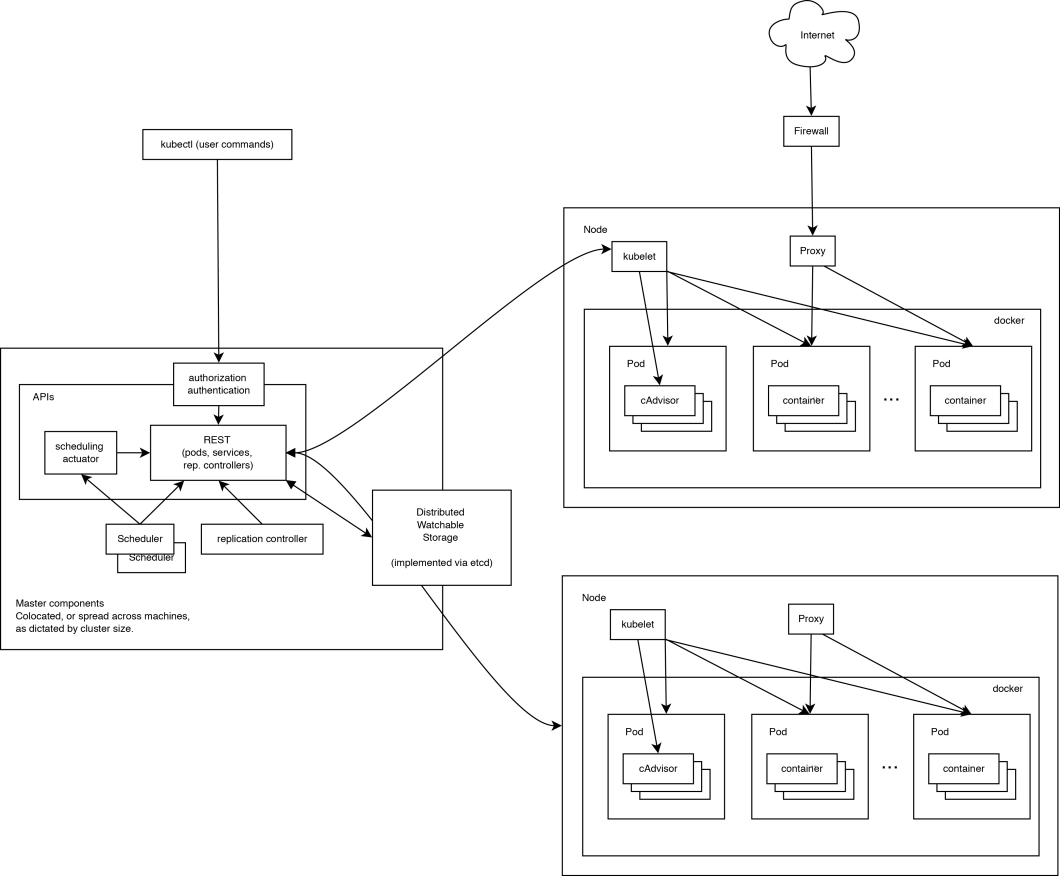
（2）以集群的方式运行、管理跨机器的容器。

（3）解决Docker跨机器容器之间的通讯问题。

（4）Kubernetes的自我修复机制使得容器集群总是运行在用户期望的状态。

当前Kubernetes支持GCE、vShpere、CoreOS、OpenShift、Azure等平台，除此之外，也可以直接运行在物理机上。

2.k8s中的关键概念



k8s的架构设计如上图所示，其关键概念主要包括以下几个方面：  
 (1) Pod   
 Pod是Kubernetes的基本操作单元，把相关的一个或多个容器构成一个Pod，通常Pod里的容器运行相同的应用。Pod包含的容器运行在同一个Node(Host)上，看作一个统一管理单元，共享相同的volumes和network namespace/IP和Port空间。

(2)Replication Controller

Replication Controller 确保任何时候Kubernetes集群中有指定数量的pod副本(replicas)在运行， 如果少于指定数量的pod副本(replicas)，Replication Controller会启动新的Container，反之会杀死多余的以保证数量不变。Replication Controller使用预先定义的pod模板创建pods，一旦创建成功，pod 模板和创建的pods没有任何关联，可以修改pod 模板而不会对已创建pods有任何影响，也可以直接更新通过Replication Controller创建的pods。对于利用pod 模板创建的pods，Replication Controller根据label selector来关联，通过修改pods的label可以删除对应的pods。

(3) Service

Service也是Kubernetes的基本操作单元，是真实应用服务的抽象，每一个服务后面都有很多对应的容器来支持，通过Proxy的port和服务selector决定服务请求传递给后端提供服务的容器，对外表现为一个单一访问接口，外部不需要了解后端如何运行，这给扩展或维护后端带来很大的好处。

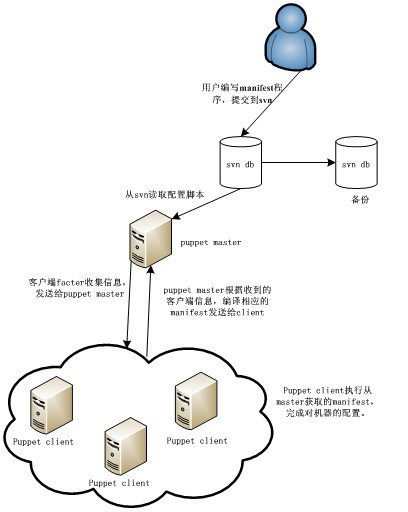
(4) Label

Label是用于区分Pod、Service、Replication Controller的key/value键值对，Pod、Service、 Replication Controller可以有多个label，但是每个label的key只能对应一个value。Labels是Service和Replication Controller运行的基础，为了将访问Service的请求转发给后端提供服务的多个容器，正是通过标识容器的labels来选择正确的容器。同样，Replication Controller也使用labels来管理通过pod 模板创建的一组容器，这样Replication Controller可以更加容易，方便地管理多个容器。

二. 使用puppet对容器做配置管理

1.puppet介绍

puppet是一个开源的软件自动化配置和部署工具，它使用简单且功能强大，正得到了越来越多地关注，现在很多大型IT公司均在使用puppet对集群中的软件进行管理和部署，puppet是基于c/s架构的。服务器端保存着所有对客户端服务器的配置代码，在puppet里面叫做manifest. 客户端下载manifest之后，可以根据manifest对服务器进行配置，例如软件包管理，用户管理和文件管理等等。



如上图所示，puppet的工作流程如下：（1）客户端puppetd调用facter，facter探测出主机的一些变量，例如主机名，内存大小，ip地址等。pupppetd 把这些信息通过ssl连接发送到服务器端； （2）服务器端的puppetmaster 检测客户端的主机名，然后找到manifest里面对应的node配置， 并对该部分内容进行解析，facter送过来的信息可以作为变量处理，node牵涉到的代码才解析，其他没牵涉的代码不解析。解析分为几个阶段，语法检查，如果语法错误就报错。如果语法没错，就继续解析，解析的结果生成一个中间的“伪代码”，然后把伪代码发给客户端；（3）客户端接收到“伪代码”，并且执行，客户端把执行结果发送给服务器；（4）服务器端把客户端的执行结果写入日志。

puppet工作过程中有两点值得注意，第一，为了保证安全，client和master之间是基于ssl和证书的，只有经master证书认证的client可以与master通信；第二，puppet会让系统保持在你所期望的某种状态并一直维持下去，如检测某个文件并保证其一直存在，保证ssh服务始终开启，如果文件被删除了或者ssh服务被关闭了，puppet下次执行时，会重新创建该文件或者启动ssh服务。

三. 使用nagios监控

1.nagios介绍

Nagios是一款开源的电脑系统和网络监视工具，能有效监控Windows、Linux和Unix的主机状态，交换机路由器等网络设置，打印机等。在系统或服务状态异常时发出邮件或短信报警第一时间通知网站运维人员，在状态恢复后发出正常的邮件或短信通知。Nagios主要有以下功能特性：

（1）监控本机及远程主机资源（CPU 负荷、磁盘利用率、进程 等）；

（2）允许用户编写自己的插件来监控特定的服务，方便地扩展自己服务的检测方法，支持多种开发语言（Shell、Perl、Python、PHP 等）

（3）具备定义网络分层结构的能力，用"parent"主机定义来表达网络主机间的关系，这种关系可被用来发现和明晰主机宕机或不可达状态；

（4）当服务或主机问题产生与解决时将告警发送给联系人（通过 EMail、短信、用户定义方式）；

（5）可以支持并实现对主机的冗余监控；

（6）可用 WEB 界面用于查看当前的网络状态、通知和故障历史、日志文件等；

（7）监控网络服务（SMTP、POP3、HTTP、FTP、PING 等）；

从实现原理的角度来讲，Nagios 软件需安装在一台独立的服务器上运行，这台服务器称为监控中心，监控中心服务器可以采用 Linux 或 Unix 操作系统；每一台被监视的硬件主机或服务都运行一个与监控中心服务器进行通信的 Nagios 软件后台程序，也可以理解为 Agent 或插件均可。监控中心服务器读取配置文件中的指令与远程的守护程序进行通信，并且指示远程的守护程序进行必要的检查。虽然 Nagios 软件必须在 Linux 或 Unix 操作系统上运行，但是远程被监控的机器可以是任何能够与其进行通信的主机，根据远程主机返回的应答，Naigos 将依据配置进行回应；接着 Nagios 将通过本地的机器进行测试，如果检测返回值不正确，Nagios 将通过一种或多种方式报警；具体原理如下图所示：

